InforMik

MAGAZYN KOMPUTEROWY „MŁODEGO TECHNIKA"



Nasz komentarz

CZAS REWANŻU: KONIUNKTURA DLA KNOW-HOW

Rozmowy z młodymi, inteligentnymi i pracującymi nad swym intelektualnym rozwojem ludźmi nader często ujawniają ich głęboką frustrację powszechnym u nas systemem wartościowania owoców pracy umysłowej lub — jak kto woli — brakiem takiego systemu. Parafrazując znane przysłowie: pokaż mi twego idola, a powiem Ci, kim jesteś. Idolami dużej części młodego pokolenia są zaś obecnie piłkarze i bożyszcza estrady.

Czym skorupka za młodu... Skoro dziś Janek wierzy, że nie ten najmądrzejszy, co ma olej w głowie, lecz ten, kto najmocniej kopie piłkę lub gromko wali w bęben albo szarpie struny gitary, to jutro Jan zlekceważy mozolny trud „jajogłowych", ślęczących przy rysownicach lub ekranach komputerów. Krociowe zarobki graczy n-tej ligi znajdują u nas aplauz kibiców, lecz zwiększone dochody twórczych inżynierów budzą małostkową zawiść. Szkopuł w tym, że kształt naszej cywilizacji tworzą właśnie jajogłowi, a nie piosenkarze ani lewi skrzydłowi. W krajach przodujących w technologii prestiż twórców techniki jest jednak bardzo wysoki, a ich znaczne dochody — uważane za rzecz normalną i właściwą.

Do czego prowadzi niedocenianie autentycznych kwalifikacji umysłowych, zwłaszcza w dziedzinie techniki, łatwo spostrzec, rozglądając się wokoło. Uświadomił nam to zwłaszcza powiew rewolucji technologicznej, który dotarł do nas w postaci gorączki mikrokomputerowej. Rewolucja informatyczna stawia naszemu społeczeństwu wyzwanie. Aby mu sprostać, potrzebna będzie elita fachowców o najwyższych kwalifikacjach i autentycznej pasji tworzenia, ciesząca się w dodatku ogólnym uznaniem.

Z informatyką zetknie się spora część społeczeństwa, zgoda. Komputery są narzędziem dla każdego — też zgoda. Większość osób stykających się z komputerem pozostanie jednak skazana na rolę biernych użytkowników, konsumentów nowej technologii, nieraz tylko zwyczajnych telegraczy. Fascynujący świat tworzenia nowych jakości, wspaniałe uczucie pełni władzy nad maszyną i świadomość własnej wiedzy i kompetencji będą zarezerwowane dla węższej grupy. W jej skład wejdą ci, co wbrew powszechnemu mniemaniu o wyższości sprzątaczki nad docentem (potwierdzanemu przez zarobki) uparcie pompowali i pompują wiedzę w swe zwoje mózgowe i zaprawiają się po nocach w rozwiązywaniu technicznych i naukowych łamigłówek. Przywilej wstępu do najtajniejszych zakamarków fascynującego świata informatyki i mikroelektroniki niech będzie dla nich zadośćuczynieniem za niedoceniany na ogół trud zdobywania kwalifikacji i ćwiczenia szarych komórek.

„InforMik" jest dedykowany szczególnie tym, którzy ze sprawności umysłowej i twórczej aktywności w dziedzinie informatyki zamierzają uczynić swą receptę na życie. InforMik zamierza przekazywać przede wszystkim tzw. *know-how*. *Know-how* zaś (w dosłownym przekładzie: wiedzieć jak) to mniej więcej tyle, co wiedza fachowa powiązana z umiejętnością praktycznego jej zastosowania. Pragniemy dostarczać pożywkę tym, którzy w komputerze widzą zarówno warsztat pracy, jak i pole do twórczego wyżycia się.

Nie należy oczekiwać od „InforMika" barwnych map gier ani wskazówek, którą piszczelą kościotrupa znalezionego pod krzemowym głazem w zaczarowanym lesie amoku należy zabić wiedźmę, aby przejść do następnej komnaty w grze „Wielkich Krańh". Ze wstydem przyznajemy, że na grach się nie znamy, a gdyby nawet, to i tak nie odważylibyśmy się konkurować z innymi czasopismami, które w międzyczasie zdobyły w dziedzinie propagacji gier zasłużoną renomę. Znajdą się natomiast na naszych łamach wskazówki, jak „rozgryzać" komputer i wykorzystywać jego ukryte możliwości, artykuły problemowe analizujące funkcjonowanie *hardware'u* i *software'u*, schematy interfejsów i innych użytecznych przystawek do samodzielnego wykonania, kursy języków i metod programowania, opisy urządzeń oraz sprawozdania z testów sprzętu i programów, a także recenzje książek fachowych. W jednym zdaniu: o tym, jak zmajstrować i podłączyć joystick — tak, jak go używać — nie. Postaramy się wyczerpująco informować o nowych, przełomowych technologiach. Zamierzamy z biegiem czasu poświęcać więcej miejsca praktycznym zastosowaniom, choć rozumiemy skromne możliwości sprzętu amatorskiego. Sensownym wykorzystaniem prostych mikrokomputerów jest edukacja, więc rozpoczynamy od działu „Komputer w szkole".

Kierując się zasadą największej użyteczności, zamierzamy poświęcić szczególną uwagę systemom stanowiącym pewne standardy — krajowe lub światowe. Z początku będą to: ZX Spectrum i ELWRO Junior oraz C-64/128, zaś spośród sprzętu bardziej zaawansowanego — IBM-PC. Dlaczego ten ostatni, piszemy na dalszych stronicach.

„InforMik" jest dla Was, drodzy Czytelnicy. Jesteśmy więc żywotnie zainteresowani Waszymi opiniami o tym wydaniu „Młodego Technika". Piszcie do nas, sygnalizujcie mankamenty, sugerujcie interesującą Was tematykę. Wasze listy będą skrupulatnie analizowane, a formuła i treść „InforMika" na bieżąco dopasowywane do Waszego zapotrzebowania. Każdy z Was ma też szansę zostać autorem „InforMika". Dla ciekawych propozycji zawsze znajdziemy miejsce na łamach. Zaznaczmy tylko, że nie interesują nas suche listingi programów, lecz raczej artykuły o charakterze problemowym, w których listingi są pożądaną ilustracją, lecz nie zasadniczą treścią. Zapraszamy do wspólnego kształtowania koniunktury dla *know-how*. Dość ględzenia — proponujemy prosty algorytm:

```
REPEAT następna stronica;
IF     interesująca
THEN   przeczytaj i wykorzystaj
ELSE   pomiń i poskarż się redakcji
UNTIL  ostatnia stronica.
```

Roland Wacławek

CO DOLEGA TWOJEMU ZX SPECTRUM!

Mikrokomputery ZX Spectrum egzystują u nas już dobre kilka lat, nie szczędząc nam przy tym mniej lub więcej kłopotliwych awarii. Uszkodzeniu ulegają pamięci, inne elementy półprzewodnikowe, niszczy się też klawiatura. Chcielibyśmy w naszym piśmie wyjść naprzeciw posiadaczom ZX Spectrum i pomóc im nieco w złagodzeniu skutków awarii czy zużycia tego bądź co bądź najpopularniejszego w Polsce mikrokomputera. Do tego jednak są nam potrzebne wiarygodne informacje na temat rodzajów i częstości występowania różnych uszkodzeń ZX Spectrum. Apelujemy zatem do wszystkich posiadaczy tego sprzętu o nadesłanie pod adresem redakcji informacji na następujące tematy:

— Jak długo używasz swoje Spectrum?

— Jakie miałeś do tej pory uszkodzenia i awarie?

— Gdzie szukałeś możliwości naprawy i czy jesteś zadowolony z jej jakości?

— Co chciałbyś w swoim komputerze poprawić, udoskonalić, zmienić itp.?

— Jakich informacji w zakresie modernizacji i serwisu Spectrum oczekujesz od redakcji?

Nie określamy absolutnie terminu nadsyłania listów na ten temat — wdzięczni będziemy za bieżące informowanie nas o aktualnych potrzebach w zakresie serwisu i modernizacji ZX Spectrum.

Szanowni Czytelnicy!

W numerze 6/86 w dodatku „InforMik" ogłosiliśmy konkurs dotyczący opracowania szczegółów konstrukcyjnych digitalizera. Niestety, NIE NADESZŁO ANI JEDNO ROZWIĄZANIE!

Przedłużamy więc termin nadsyłania rozwiązań **do sierpnia 1987**, jednocześnie przypominając, że główną nagrodą jest dwubiegowa wiertarka CELMA-BOSCH, ufundowana przez Centralną Składnicę Harcerską.

Proponujemy naszym Czytelnikom opracowanie szczegółów konstrukcyjnych digitalizera (konstrukcja mechaniczna, dobór elementów itp.) oraz odpowiedniego programu sterującego, umożliwiającego szybkie przetwarzanie położenia pisaka na współrzędne (x, y), a także na graficzne przeniesienie wyników digitalizacji na ekran komputera.

Całość winna zapewniać dokładność i rozdzielczość zbliżoną do osiągalnej w przypadku standardowej grafiki ZX Spectrum (głównym ogranicznikiem będzie błąd liniowości potencjometrów) jak również szybkość pozwalająca na swobodne, niezbyt szybkie prowadzenie pisaka po rysunku.

Wskazane jest zatem napisanie przynajmniej części sterującej w języku wewnętrznym mikroprocesora (program sterujący i odmierzający czas nie może być umieszczony w „dolnym" 16 KB pamięci!); poza tym nie wnosimy żadnych ograniczeń.

młody TECHNIK

InforMik

MAGAZYN KOMPUTEROWY
„MŁODEGO TECHNIKA"
NR 1. ROCZNIK I.

SPIS TREŚCI

ARTYKUŁY:
Magia trzech liter — Roland Wacławek 2
SCHNEIDER PC: Zatarte granice (rw) 8
Pamięci stałe — Krzysztof Wiśniewski 10
Komputerowe przypominadło — Tadeusz Basista 13
Jak z ZX Spectrum zrobić komputer cz. 1 — Grzegorz Zalot 18

FELIETONY:
Czas rewanżu: koniunktura dla know-how (rw) . . . . . II str. okł.
Jeden pasterz — Jerzy Klawiński 1

DZIAŁY:
SEMINARIUM INFORMIKA: Asembler Gens 3 cz. I — Tadeusz Basista 28
KOMPUTER W SZKOLE: Jak zrobić komputerowy test — Marek T. Szczepański 30
CIEKAWE KSIĄŻKI 12

INNE:
Nasz test: TIMEX 2048 — Roland Wacławek 26

OKŁADKA:
I — Tak wyobrażamy sobie posiedzenie Rady Redakcyjnej „InforMika" w roku 2103
III—IV — Stacja dysków elastycznych — Piotr Postawka

Numer ilustrował: Jerzy Flisak
Zdjęcia w numerze: W. P. Jabłoński, Grzegorz Zalot, ze zbiorów redakcji

JEDEN PASTERZ



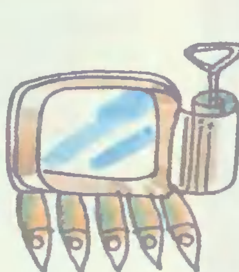
Rozkładając papuzio barwne stronice zachodnich magazynów komputerowych, pełnych reklam zarówno dyskretnych, jak i krzykliwych, zastanawiam się, na ile wpływają one na nasz sensowny wybór stosownego dla naszych możliwości i umiejętności sprzętu. Spory pomiędzy użytkownikami wyrobów poszczególnych firm dowodzą, jak źle jest z naszą wiedzą informatyczną, która ogranicza się najczęściej do mniej lub bardziej gruntownej lektury broszurki zawartej w opakowaniu komputera i powtarzaniu kilku sloganów, jakimi wychwala swój sprzęt „nasz" producent dyskredytując jednocześnie (słynne porównania tzw. danych użytkowych) komputery konkurencji. Oczywiście fakt, że owe dane są specyficznie dobierane, nikogo nie dziwi (lub nie powinno dziwić!).

Smutne to, bo najdoskonalszy nawet z rodziny IBM-ów w rękach profana jest tak samo bezużyteczny, jak pusta puszka po sardynkach. Fakt posiadania tego, a nie innego typu komputera świadczy najczęściej o statusie majątkowym właściciela, a nie o jego umiejętnościach. Smutne, bo czasem ludzi zdolnych i pracowitych nie stać na sprzęt lepszy niż Spectrum...

Czy można w tej sytuacji reklamować jakikolwiek sprzęt jako najlepszy, jedyny i uniwersalny? Czyż nie lepiej dać ludziom do ręki narzędzie do pracy na każdym z istniejących typów komputerów? Dlaczego nie pokazać im, jak mogą pracą i wiedzą poprawić własny skromny „park" komputerowy?

Temu wszystkiemu ma służyć „InforMik", gdyż jego redakcja pragnie pomóc tym, którzy w swym samokształceniu zderzyli się z barierą niemożności sprzętowej i braku wiedzy. Nie będziemy dążyć do „zmonopolizowania" rządu dusz wszystkich komputerowych hobbistów, jak również postaramy się uniknąć wkraczania w kompetencje specjalistów — mamy też służyć tym wszystkim, którzy chcą się uczyć jak być sprawnym użytkownikiem komputera, a nie jaki mieć komputer. We współczesnej technice czasy monopolistów raczej się skończyły i biblijna „jedna owczarnia i jeden pasterz" nie mają w tej sferze racji bytu. Czytajcie więc, Drodzy Czytelnicy, porównujcie, wybierajcie — macie w czym. A jeśli na stałe wybierzecie jako swą lekturę „InforMik", to na pewno nie dlatego, że jego redakcja będzie się reklamowała jako „jedynie słuszny" pasterz...

Jerzy Klawiński









MAGIA TRZECH LITER

Roland Wacławek

Choć pierwsze komputery osobiste: PET i Apple już w roku 1977 zrobiły prawdziwą karierę, koncern IBM długo ignorował informatyczne „drobnoustroje". Trudno się dziwić, skoro źródłem dotychczasowych sukcesów firmy były głównie duże systemy komputerowe zamknięte w obowiązkowo błękitnych obudowach (stąd żartobliwe określenie IBM: *Big Blue*). Prestiż IBM opierał się na kompleksowości usług, dobrej jakości wyrobów oraz sprawności gęstej sieci serwisowej. Dopiero w roku 1981 IBM zaprezentowała własny komputer osobisty IBM-PC. W ciągu dwóch lat model ten zdominował rynek sprzętu profesjonalnego w Ameryce i zaczął szturmować inne kontynenty — głównie Europę. Powstał powszechnie przyjęty standard, dla którego tworzono masowo wartościowe oprogramowanie i osprzęt. Oprócz Apple konkurenci IBM zrezygnowali z lansowania własnych systemów i przestawili się na produkcję komputerów tzw. kompatybilnych, czyli zgodnych ze standardem IBM-PC. Dziś, po prawie sześciu latach, pozycja IBM-PC utwierdziła się jeszcze mocniej — szybko rozpowszechnia się znacznie wydajniejszy komputer IBM-AT, wywodzący się od PC i akceptujący większość jego oprogramowania.

Oryginalny IBM-PC — protoplasta licznej rodziny komputerów kompatybilnych



Co sprawiło, że IBM-PC podbił rynek i ugruntował nieoficjalny, lecz obowiązujący standard? Prestiż IBM? Z pewnością, ale nie tylko. Komputer nie jest wcale rewelacją techniczną, ba — nie był nią nawet w roku 1981, jego konstrukcja jest jednak solidna i opiera się na standardowych i ogólnie dostępnych układach scalonych. IBM udostępnia też pełną dokumentację techniczną mikrokomputera (ze skomentowanym listingiem BIOS-ROM włącznie) i zastosowanego w nim systemu operacyjnego PC-DOS. W tej sytuacji kopiować komputer może każdy, kto tylko ma ochotę. Zastrzeżone są jedynie: zawartość ROM oraz trzy magiczne litery na obudowie...

Podczas gdy firma Apple zażarcie procesowała się ze wszystkimi, którzy ośmielili się wytwarzać wyroby naśladujące oryginalne „jabłuszka", IBM zdawała się nie dostrzegać wzbierającej fali komputerów kompatybilnych, czasem wiernych kopii IBM-PC, i nie czyniła konkurentom większych wstrętów. Ta pozornie pasywna strategia okazała się jednak dalekowzroczna i na dłuższą metę — skuteczna. Obfitość komputerów akceptujących to samo oprogramowanie i korzystających z tego samego osprzętu, oraz urządzeń peryferyjnych do nich, stworzyła warunki do długo oczekiwanej stabilizacji rynku mikrokomputerowego. Fakt, że najpierw amerykańscy (np. Compaq), a potem azjatyccy producenci oferowali masowo sprzęt tańszy i zarazem ulepszony w porównaniu z oryginalnym, odebrał co prawda IBM część doraźnych zysków, ale i zapewnił długofalowe korzyści. Żaden z naśladowców, reklamując swój produkt, nie omieszkał bowiem podkreślić jego zgodności ze standardem IBM. Z pomocą konkurentów koncern IBM szybko zuniformizował podaż, odbierając klientelę krnąbrnym wytwórcom „egzotycznych", tzn. niezgodnych z IBM-PC systemów.

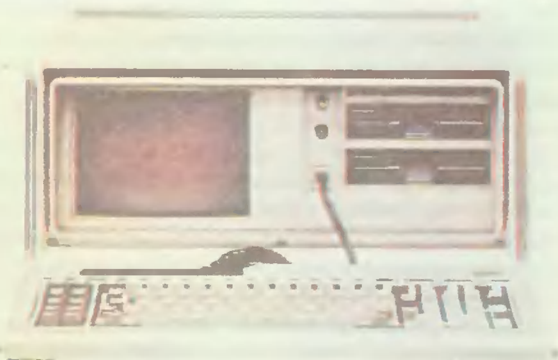
Wadą pierwotnej wersji IBM-PC był słaby zasilacz sieciowy (63 W), dostający „zadyszki" po instalacji

twardego dysku. Pierwsze rozwiązanie, w postaci dodatkowego pudła zawierającego dysk i własny zasilacz, nie zdało egzaminu jako drogie i nieporęczne. Dlatego w roku 1983 IBM przedstawiła mocniejszy zasilacz (130 W). Otrzymał on większą pamięć RAM i obok jednej stacji dyskietek — dysk twardy 10 MB. Przeprojektowano też płytę montażową, powiększając liczbę gniazd dla dodatkowych pakietów z 5 do 8. W zasadzie XT był jednak tylko ulepszonym PC, zachowując z nim pełną zgodność programową. W roku 1984 pojawił się nowy model: IBM-PC/AT, oparty o nowy, znacznie wydajniejszy mikroprocesor Intel-80286. Model ten może korzystać ze znacznej większości oprogramowania IBM-PC/XT.

Silna konkurencja doprowadziła do załamania cen komputerów kompatybilnych z IBM-PC i sprowadziła je do poziomu zbliżonego do cen typowych komputerów domowych. Czy wobec nowej fali technologicznej, reprezentowanej przez Atari ST lub Amigę, jest jednak sens uważać IBM-PC/XT za komputer domowy? Wszystko zależy od wymagań i sposobu podejścia. Kto kładzie nacisk na wyrafinowane możliwości sprzętowe, zwłaszcza graficzne i muzyczne, niech wybierze któryś z tamtych systemów. Żądając w pierwszym rzędzie elastyczności i użyteczności, warto postawić na IBM-PC/XT.

O użyteczności komputera decyduje nie tylko wyrafinowany osprzęt, lecz — może nawet w większym jeszcze stopniu — dobre oprogramowanie. IBM-PC nie ma tu w swej klasie liczącej się konkurencji. Firmy softwarowe chętnie tworzą oprogramowanie dla IBM-PC, mając na uwadze szeroki rynek i zamożną klientelę. Program eksploatowany w twardych warunkach profesjonalnych musi oczywiście sprostać innym wymogom jakościowym niż software dla zabawkowych: Atari XL lub Spectrum, IBM-PC ma zatem dziś oprogramowanie nie tylko najbogatsze ilościowo, lecz i najwartościowsze, najlepiej przetestowane i udokumentowane. Bogata jest literatura fachowa, poczynając od podręczników systemowych, a kończąc na poradnikach posługiwania się co popularniejszymi programami, jak dBase, Framework, itd. Oprogramowanie jest coraz dostępniejsze także w Polsce, tym bardziej że oprócz oryginalnych polskich programów pojawiają się coraz liczniejsze adaptacje i spolszczenia znanych programów zagranicznych. Dostępność software będzie rosła, gdyż IBM-PC/XT staje się standardem także w Polsce, a do produkcji trafiają rodzime (?) systemy kompatybilne, np. Mazovia. Znając operatywność amatorów informatyki, można przypuszczać, że prywatni użytkownicy IBM-PC w Polsce będą

PC-Portable — przenośna wersja IBM-PC





Komputer kompatybilny z IBM-PC rodem z Tajwanu w konfiguracji minimalnej. Naśladownictwo posunięto nawet do obudowy i do układu klawiatury

pławić się w obfitości wysokiej klasy oprogramowania, budząc zazdrość zwolenników innych systemów. Jednym słowem, użytkownicy IBM-PC mogą liczyć na wiele przywilejów.

Podstawowym systemem operacyjnym IBM-PC jest PC-DOS, praktycznie zgodny z MS-DOS, lansowanym przez firmę MICROSOFT. W systemach tych pracuje większość oprogramowania. Znacznie mniejsze znaczenie ma CP/M-86, autorstwa Digital Research. Wszystko to są systemy jednozadaniowe. W przypadku większych wymagań, albo tam, gdzie trzeba zorganizować np. sieć mikrokomputerów lub pracę wielodostępną z udziałem kilku dodatkowych końcówek, można użyć innych systemów, np. Concurrent CP/M (Digital Research) lub XENIX (ten ostatni będący mutacją UNIX-a, opracowaną przez MICROSOFT).

W odróżnieniu od Atari ST lub Amigi, stanowiących raczej zamknięte systemy, IBM-PC/XT jest układanką z klocków. W razie potrzeby można mnożyć interfejsy i pamięci masowe, wstawiać do środka specjalistyczne pakiety, jednym słowem — dopasować komputer do indywidualnych wymagań. Atari ST lub Amiga, nowoczesne w chwili zakupu, będą się już tylko moralnie starzeć. Tymczasem IBM-PC/XT pozwala się stopniowo modernizować, wymieniając poszczególne elementy. Mówi się np., że Atari ST góruje nad IBM-PC grafiką. Jeśli rozważymy standardowy pakiet graficzny CGA (*Color Graphics Adapter*) — zgoda. Gdy jednak wymienimy CGA na EGA (*Enhanced Graphics Adapter*), Atari ST zostanie daleko w tyle — i to bez szansy rewanżu. Procesor Atari ST jest wydajniejszy — zgoda. Kod maszynowy MC 68000 jest jednak mniej oszczędny niż 8088. Turbo-mainboard zmniejsza znacznie przewagę Atari, zaś koprocesor arytmetyczny (niemożliwy do instalacji w Atari ST) daje IBM-PC znaczną przewagę w zastosowaniach numerycznych i wielu programach graficznych, wymagających licznych operacji trygonometrycznych. Atari ST reprezentuje filozofię komputera domowego, taniego w wersji podstawowej, lecz trudnego i kosztownego w rozbudowie. Kwestia ceny też jest dyskusyjna. Z uwagi na specjalizowane układy scalone, zawarte w Atari ST lub Amidze, nie można





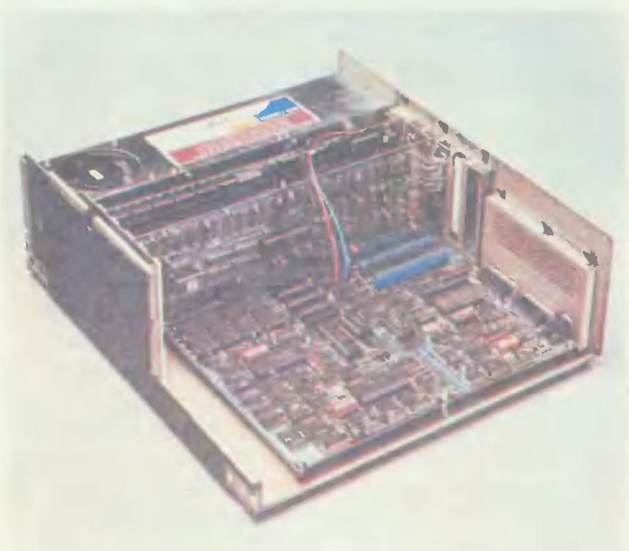
Różnice między większością komputerów kompatybilnych są niewielkie i dotyczą głównie aparycji oraz szczegółów konfiguracyjnych

mieć nadziei na konkurencję cenową ze strony niezależnych wytwórców. Przyzwoity komputer klasy IBM-PC można nabyć poniżej ceny ATARI 520 ST ze stacją dysków.

Masowość produkcji osprzętu dla IBM-PC/XT i silna konkurencja na rynku sprawiają, że dodatkowe stacje dysków, drukarki, interfejsy itd. są dostępne w szerokim wyborze i częściowo po dość atrakcyjnych cenach (dotyczy to zwłaszcza starszych modeli — tych sprzed roku). Przy rozumnym planowaniu zakupów początkowa większa inwestycja w podstawową konfigurację IBM-PC/XT umożliwi z czasem osiągnięcie wymiernych oszczędności.

Rozwinięta sieć serwisowo-usługowa oraz renoma IBM są magnesem dla licznej klienteli, zwłaszcza wśród wielkich przedsiębiorstw. Słone ceny wyrobów IBM równoważy gwarancja stabilności i szybkiej pomocy w razie kłopotów. Nabywca indywidualny chętnie sięga natomiast po tańsze wyroby. W polskich warunkach trzy magiczne litery nie dają użytkownikowi, zwłaszcza prywatnemu, istotnych korzyści. W tej sytuacji podstawowym kryterium przy zakupie jest jak najniższa cena sprzętu, swoboda wyboru konfiguracji odpowiedniej do własnych potrzeb i brak narzutów za „przymusowo" dołączane oprogramowanie. Warunki te najlepiej spełniają komputery składane z pakietów wytwarzanych w Azji Południowo-Wschodniej, np. na Tajwanie. IBM-PC w minimalnej konfiguracji można dziś nabyć prawie za tą samą cenę, co przed dwoma laty Commodore C-64 ze stacją dysków VC-1541.

Rzut oka do wnętrza komputera kompatybilnego. Ten egzemplarz nie jest niewolniczą kopią IBM-PC



Na antypodach ceny są zachęcające, ale koszt transportu komputera i związane z tym ryzyko — wysokie. W razie ewentualnych braków i usterek możliwości reklamacji praktycznie są prawie żadne, nie brak więc osób, które na sprowadzaniu komputerów wprost z Tajwanu nieźle się nacięły. Na szczęście sprzęt tajwański sprowadzają hurtowo także liczne firmy zachodnioeuropejskie i sprzedają częściowo pod własną marką. Niektóre z tych przedsiębiorstw uprawiają sprzedaż wysyłkową do Polski. Sprzęt nabywany w Europie Zachodniej cieszy się lepszą opinią, chociaż pozostaje problem realizacji uprawnień gwarancyjnych. Ostatnio dotarła do nas jednak wiadomość o firmie z RFN, dostarczającej do Polski komputery, w tym klasy IBM-PC, na całkiem innych zasadach. Otóż klient odbiera zamówiony sprzęt w Polsce i ma prawo zażądać jego wstępnego przeglądu. Przy okazji może zaopatrzyć się w oprogramowanie i niezbędną dokumentację oraz literaturę w języku polskim. Oprócz tego sprowadzane komputery objęte są gwarancją realizowaną w Polsce przez wyspecjalizowany zakład, świadczący na życzenie także usługi instalacyjne i adaptacyjne oraz serwis pogwarancyjny. Wypada tylko wyrazić żal, że tak atrakcyjny sposób zaopatrywania się w sprzęt komputerowy jest dostępny jedynie dla posiadaczy kont dewizowych lub krewnych za granicą.

Jak jest skonstruowany komputer klasy IBM-PC/XT? Rzućmy okiem do wnętrza. Prawy, tylny róg obudowy zajmuje zamknięty w blaszanym pudle zasilacz sieciowy, zwykle o mocy 135 lub 190 W. Jest to zasilacz impulsowy dzięki czemu zapewnia on niezbędną izolację galwaniczną bez użycia ciężkiego transformatora sieciowego. Zasilacz jest zabezpieczony przed zwarciem i dostarcza następujących napięć: +5 V (do 15 A w wersji 135 W), +12 V (do 4,5 A), −5 V i −12 V (do 0,5 A). Układy cyfrowe konsumują prawie wyłącznie napięcie +5 V, +12 V zasila stacje dysków twardych i elastycznych oraz — wraz z napięciem −12 V — interfejs RS232. Źródło −5 V to relikt z niemowlęcych lat PC, gdy stosowano jeszcze pamięci RAM typu 4116, wymagające ujemnej polaryzacji. W zasilacz wbudowany jest wentylator, który nie tylko chłodzi jego wnętrze, ale wysysając powietrze z pudła komputera zapewnia intensywną cyrkulację powietrza i dobre odprowadzanie ciepła ze wszystkich podzespołów. W praktyce zasilacz ma spore rezerwy mocy, można więc zasilać z niego wiele dodatkowych pakietów i przystawek.

Lewą połowę obudowy zajmuje pakiet podstawowy (płyta główna), czyli tzw. *mainboard* (czytaj: majnbord). Mieści on kompletny system mikroprocesorowy z procesorem Intel-8088. Obok procesora znajduje się obowiązkowa podstawka dla koprocesora arytmetycznego Intel-8087, wyspecjalizowanego w szybkich operacjach na liczbach o dużej precyzji, zwłaszcza w formacie wykładniczym. W zastosowaniach numerycznych zwiększa on wydajność IBM-PC przynajmniej o rząd. Aby zainstalować 8087, wystarczy włożyć go w podstawkę i przestawić przełącznik na płycie głównej.

W oryginalnym IBM-PC częstotliwość zegarowa procesora wynosi 4.77 MHz. Wśród systemów kompatybilnych rozpowszechnił się tzw. *Turbo-mainboard*, pozwalający przełączać częstotliwość zegarową między 4.77, a 8 (niekiedy nawet 10) MHz. Zegar 8 MHz oznacza 1.67 raza większą wydajność procesora niż w oryginale. Częstotliwość 4.77 MHz jest jednak wymagana przez niektóre programy ściśle współdziałające ze sprzętem, np. przez część programów kopiujących. *Turbo-mainboard* posiada procesor 8088-2, zaś jako koprocesora należy użyć wersji 8087-2, z wyższą dopuszczalną częstotliwością zegarową.

Oprócz procesora, *mainboard* zawiera pamięć operacyjną ROM i RAM. Najważniejszą i obowiązkową częścią ROM jest liczący 8 KB tzw. BIOS (*Basic Input/Output System*). Wbrew pozorom, nie ma on nic wspólnego z BASIC-em, lecz stanowi głównie zbiór procedur bezpośrednio współdziałających ze sprzętem. BIOS zawiera m.in.: test pamięci RAM, wykonywany automatycznie po włączeniu komputera, handlery (procedury obsługi) urządzeń wejścia/wyjścia i usługowe programy systemowe, dostarczające np. informacji o stanie systemu. Przykładem mogą być podprogramy odczytu lub zapisu bloku danych na dyskietkę, wczytania znaku z klawiatury lub wyprowadzenia go na ekran lub drukarkę. BIOS zawiera też wzorce znaków ekranowych o kodach 0..127, wykorzystywane przy pracy w trybie graficznym. BIOS IBM-PC objęty jest patentem, lecz producenci sprzętu kompatybilnego omijają to ograniczenie, realizując funkcje BIOS w inny sposób.

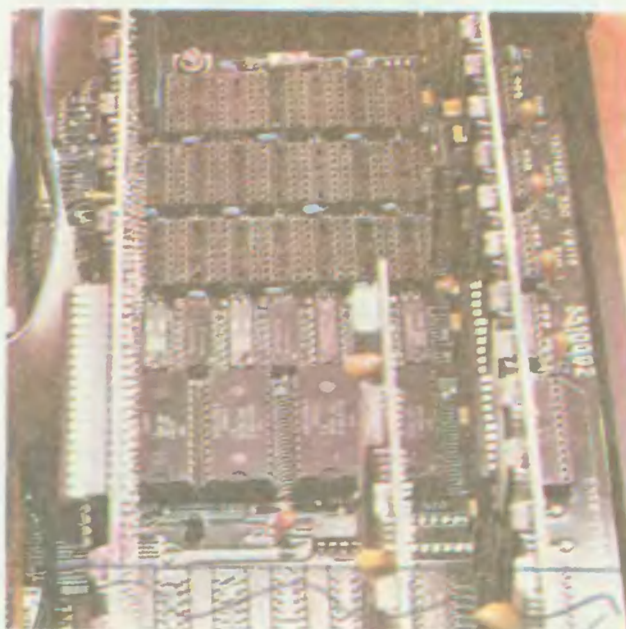


Jednostka twardego dysku typu Winchester, przeznaczona do zabudowy w IBM-PC

Oryginalny PC zawiera dodatkowe 32 KB zastrzeżonej pamięci ROM z interpreterem języka BASIC. Jest on uruchamiany automatycznie po włączeniu komputera, jeśli nie można załadować z dysku systemu operacyjnego. Większość komputerów kompatybilnych zamiast układów BASIC-ROM ma tylko cztery puste podstawki. Niewielka to strata, gdyż można załadować z dyskietki GW-BASIC, zgodny z BASICEM IBM (tzw. BASICA).

W IBM-PC pamięć RAM liczyła 64 KB i składała się z czterech bloków 16 KB po 9 układów typu 4116. Dziewiąty układ zawierał bit parzystości. Aby zwiększyć pewność eksploatacji, IBM-PC ma bowiem sprzętowy układ wykrywania błędów pamięci RAM, *mainboard* wersji XT mógł mieć do czterech bloków po 64 KB, złożonych z układów 4164. Nowoczesne *mainboardy*

Fragment mainboardu IBM-PC. Widać układy pamięci ROM (u dołu) oraz cztery banki pamięci RAM po 64 KB. Obsadzony układami jest tylko jeden z nich — reszta podstawek czeka na wstawienie układów scalonych





Zręczne ręce są jedynym narzędziem potrzebnym do rozbudowy pamięci RAM w większości komputerów klasy IBM-PC

kompatybilne mieszczą do 640 KB RAM: 2×256 KB z układów 41256 i 2×64 z 4164. 640 KB wyznacza praktyczną granicę rozbudowy RAM w IBM-PC. Intel-8088 może wprawdzie adresować do 1 MB pamięci operacyjnej, ale pozostała część przestrzeni adresowej została przez IBM zarezerwowana dla ROM i pamięci ekranu sterowników graficznych. Przy zakupie komputera z pamięcią mniejszą niż 640 KB warto sprawdzić, czy ma on *mainboard* przystosowany do pełnej rozbudowy RAM. Można co prawda zainstalować RAM na dodatkowych kartach, lecz oznacza to spory a niepotrzebny wydatek. Aby rozbudować RAM na *mainboardzie*, wystarczy na ogół wstawić taniutkie dziś układy scalone w przygotowane cokoły, po czym przestawić kilka zworek.

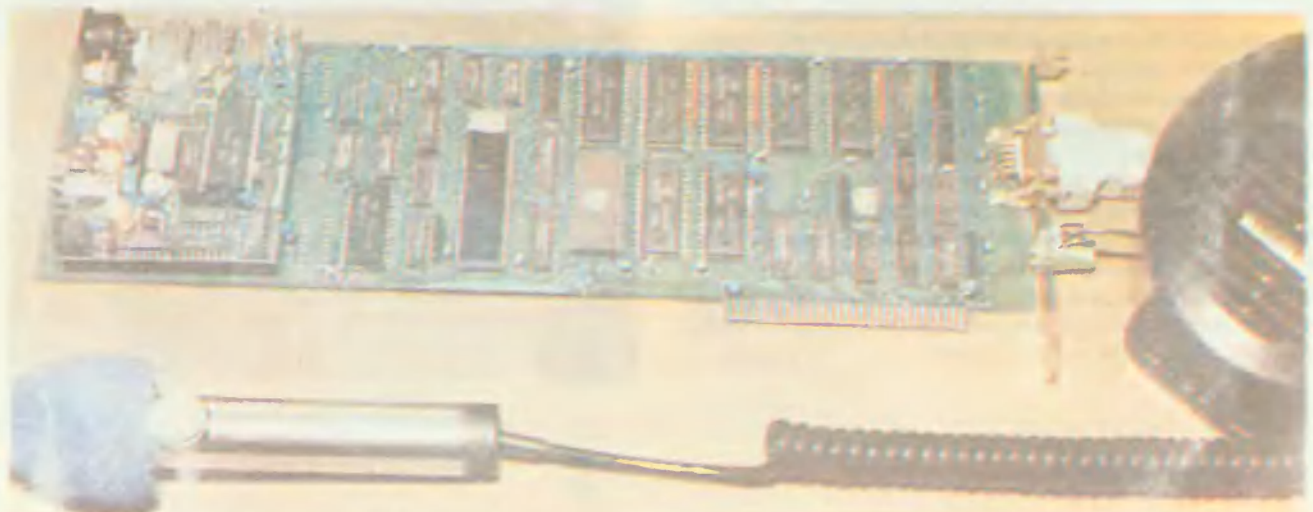
Mainboard zawiera jeszcze niezbędne obwody sterujące i układy pomocnicze oraz peryferyjne. Są to:

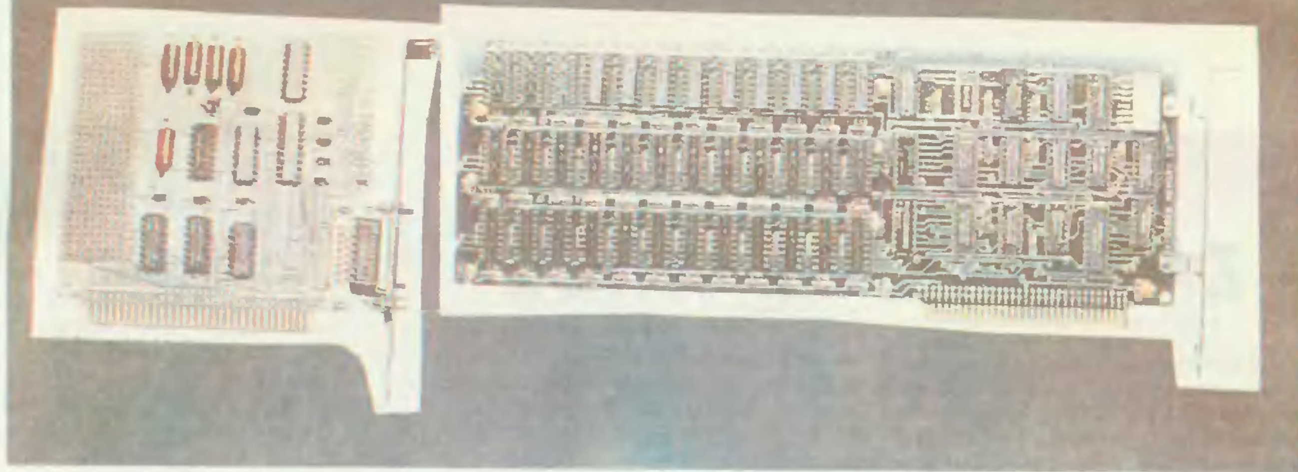
— Sterownik przerwań priorytetowych (PIC) 8259. Obsługuje 8 poziomów przerwań maskowalnych IRQ, w tym: od zegara systemowego, klawiatury, sterownika dysków i interfejsów: szeregowego i równoległego.
— Sterownik bezpośredniego dostępu do pamięci (DMAC) 8237. Spośród jego czterech kanałów jeden odświeża dynamiczne pamięci RAM, drugi pośredniczy w transmisji danych między RAM, a stacjami dysków, a dwa pozostałe są nie używane.
— Układ czasowy typu 8253, zawierający trzy programowane liczniki. Jeden z nich pełni funkcję zegara systemowego, powodując co 54.925 milisekundy przerwanie IRQ, drugi odmierza tempo odświeżania pamięci dynamicznych, zaś trzeci służy do generacji efektów akustycznych.
— Równoległy układ wejścia/wyjścia (PPI) 8255. Obsługuje interfejs klawiatury oraz głośniczek, odczytuje nastawy przełączników konfiguracyjnych na *mainboardzie*, steruje detekcją błędów parzystości.

Podłączenie klawiatury rozwiązano oryginalnie i efektywnie. Klawiatura ma własny, jednoukładowy mikrokomputer Intel-8048, 8049 lub 8749. Testuje on na bieżąco stan klawiszy i dba o właściwą interpretację przycisków SHIFT, CTRL, itd. Informacja o naciśnięciu i o zwolnieniu każdego klawisza przekazywana jest do komputera elastycznym przewodem za pośrednictwem specjalnego, synchronicznego złącza szeregowego. Każde naciśnięcie lub zwolnienie klawisza wywołuje przerwanie, główny procesor nie musi zatem na bieżąco śledzić stanu klawiatury. Klawiatura zasilana jest z komputera. Układ klawiatury daje powody do krytyki. Ważne klawisze, jak SHIFT lub ENTER, są małe i niewygodne. Brak oddzielnego bloku sterowania kursorem: używa się tu pola klawiszy numerycznych. Niestety, nie ma sygnalizacji aktualnej funkcji tego pola, podobnie jak przełączników: CAPS LOCK, NUM LOCK i SCROLL LOCK. Obecnie można nabyć jednak znacznie ulepszone, choć nieco droższe klawiatury, z powiększonymi klawiszami SHIFT, ENTER itd. i z sygnalizacją stanu. Bardzo dobre są np. klawiatury wzorowane na IBM AT, wyposażone w przełącznik XT/AT. Przy przetwarzaniu tekstów niewielka inwestycja w lepszą klawiaturę szybko się amortyzuje.

W tylnej części *mainboardu* znajduje się 5 (w przypadku PC) lub 8 (XT) standardowych gniazd magistrali systemowej (magistrala jest buforowana, zwarcia nie grożą rozległym uszkodzeniem systemu). W gniazda te (ang. *slot*) wstawia się pakiety (karty, ang. *card*) pełniące różne funkcje. Obowiązkowy jest pakiet sterownika ekranu. Spośród wielu typów w warunkach amatorskich najodpowiedniejszy jest CGA (*Color Graphics Adapter*), czyli pakiet grafiki barwnej. Ma on 16 KB niezależnej pamięci ekranu, może operować 16 barwami i pozwala pracować w 6 trybach: tekstowych — jedno- i wielobarwnym po 40 lub 80 znaków w 25 wierszach, graficznym jednobarwnym 640×200 punktów, graficznym wielobarwnym 320×200 (każdy punkt może przyjąć jedną z czterech wstępnie zdefiniowanych barw). Dostępny, lecz nie wykorzystywany jest tryb 160×100 barwnych punktów. CGA ma wyjście zarówno

Bogactwo osprzętu to poważna zaleta IBM-PC. Na zdjęciu specjalistyczny pakiet do analizy i syntezy mowy





Typowe pakiety dodatkowe do IBM-PC — karta rozszerzenia pamięci RAM (długa) i interfejs joysticków (krótka)

typu RGB, jak i Composite Video. Do tego ostatniego można dołączyć standardowy monitor, np. Neptun 156, albo nawet adaptowany, czarno-biały odbiornik OTV. Czytelność na ekranie telewizora jest zadowalająca nawet w trybie 80 symboli w wierszu, gdyż stosowane jest pogrubianie pionowych elementów znaków. Z kartą CGA może współpracować praktycznie każde oprogramowanie — w szczególności gry.

Drugi co do popularności jest monochromatyczny pakiet graficzny Hercules, wyposażony w pamięć ekranu 64 KB i dwa tryby pracy: graficzny 720 × 348 punktów i tekstowy 25 linii po 80 znaków. Krój znaków jest przyjemniejszy dla oka niż w CGA, lecz Hercules jest mniej uniwersalny i nie działa z każdym oprogramowaniem (np. symulator lotu firmy Microsoft wymaga CGA). Hercules jest natomiast doskonały przy projektowaniu itp. Pakiet ma wbudowany dodatkowy interfejs Centronics.

Obok kontrolera ekranu niezbędny jest sterownik dysków elastycznych i interfejs Centronics do podłączenia drukarki, bardzo przydaje się też interfejs RS232. Elementy te są wprawdzie dostępne oddzielnie, lecz bardziej opłaca się nabyć pakiet wielofunkcyjny. W typowych systemach optymalny jest tzw. pakiet Multi I/O, zawierający sterownik dla dwóch stacji dysków elastycznych 5,25 cala, interfejs Centronics i dwa interfejsy RS232: jeden kompletny, drugi bez układu 8250 (w razie potrzeby wystarczy go wstawić). Oprócz tego pakiet zawiera interfejs dla joysticków, ang. *game-port* (IBM-PC używa joysticków analogowych, potencjometrycznych) i zegar z zasilaniem akumulatorowym, przechowujący aktualny czas i datę. Zegar ten, często niedoceniany, jest wspaniałym udogodnieniem. System PC-DOS zapisuje np. bowiem w katalogu dyskietki czas i datę każdej modyfikacji zbioru. W krytycznych sytuacjach pozwala to ustalić aktualną wersję zbioru lub uratować dane. Czas i datę można wprawdzie podać z klawiatury po włączeniu komputera, lecz w codziennej praktyce jest to uciążliwe.

Karta wielofunkcyjna zamyka minimalną konfigurację, niezbędną w warunkach amatorskich, w warsztacie rzemieślniczym itd. Twarde dyski, karty graficzne EGA itd. zwiększają wprawdzie wydajność systemu, ale i podnoszą poważnie jego cenę. Osprzęt ten jednak szybko tanieje i można liczyć na jego dokupienie po pewnym czasie na znacznie dogodniejszych warunkach. Przydaje się jeszcze drukarka, ale tutaj wybór jest wyjątkowo bogaty. Większość producentów dba bowiem o zgodność swych modeli ze standerdem IBM.

Frontowa ściana obudowy ma zasłonięte zaślepkami otwory, w które można wstawić do 4 jednostek dysków twardych lub elastycznych. Pierwsze modele IBM-PC wyposażano w jednogłowicowe (tzn. jednostronne) stacje dysków 5,25 cala o pojemności 160 KB. Powierzchnia dysku dzieliła się na 40 ścieżek zapisanych z podwójną gęstością (DD), 8 sektorów po 512 bajtów na każdej z nich (dane te dotyczą systemu PC-DOS). Wkrótce zastosowano stacje dwustronne o podwojonej pojemności. Następnym ulepszeniem był wprowadzony od wersji PC-DOS 2.0 nowy sposób formatowania dysków — po 9 sektorów na ścieżce. Pojemność dyskietki wzrosła w ten sposób do 360 KB, przy katalogu mieszczącym 112 zbiorów. PC-DOS 2.0 umożliwił też hierarchiczną (drzewiastą) organizację katalogu zbiorów, zapożyczoną z systemu UNIX, a wygodną szczególnie w przypadku pamięci dyskowych o dużych pojemnościach.

W warunkach amatorskich wystarcza jedna stacja dysków o pojemności 360 KB: drugi dysk można zawsze zasymulować w pamięci operacyjnej (tzw. RAM-DISK). Konfiguracja z dwoma dyskami i pamięcią 640 KB otwiera już dostęp do najwartościowszego oprogramowania, jak dBASE III, Framework-II, AUTO-CAD itd. Jeśli zależy nam na systemie jak najtańszym, to granicę przyzwoitości wyznaczają: 1 stacja dysków i 256 KB RAM. Funkcjonują wtedy m.in.: WordStar 3,4 (i to nawet z małym RAM-dyskiem na nakładki), TURBO-PASCAL, GW-BASIC, Macroassembler, Sidekick, GFX, Flight Simulator II i wiele innych. W pierwszej kolejności warto jednak walczyć o rozbudowę pamięci RAM do 640 KB: znaczy ona więcej, niż dodatkowa stacja dysków, a jest znacznie tańsza. Wracając do obudowy: najpraktyczniejszy jest wariant z odchyloną górną pokrywą, mocowaną na boczne zatrzaski.

Jak widać, obecne komputery klasy IBM-PC/XT mocno różnią się od swych prekursorów i oferują lepsze parametry. Rodzina IBM-PC przeszła długą ewolucję. Wystarczy pomyśleć, że oryginalny PC współpracował z magnetofonem jako pamięcią masową (w BASIC-u) i można go było nabyć bez pamięci dyskowej, a na życzenie — z pamięcią RAM odchudzoną do zaledwie 16 KB!

W następnych numerach „InforMika" weźmiemy pod lupę tajemnice IBM-PC, tak w dziedzinie *hardware*, jak i *software*. Szczególną uwagę poświęcimy skromnym konfiguracjom, bez twardego dysku. — takim, jakie można będzie spotkać w domu, w klubie lub w szkolnej pracowni.

Schneider PC:

ZATARTE GRANICE

Poprzedzony licznymi pogłoskami i zapowiedziami, po półrocznym oczekiwaniu, pojawił się nowy mikrokomputer marki Amstrad-Schneider, oznaczony PC 1512. Zrywa on z linią ośmiobitowców serii CPC i PCW, lecz pozostaje wierny strategii producentów, stanowiąc kompletny i estetyczny system o oryginalnej aparycji. Obudowę skonstruowano nie z blachy, lecz z tworzyw sztucznych, znacznie redukując jej gabaryty w porównaniu z typowymi komputerami osobistymi.

Choć Schneider-PC powstał z myślą o indywidualnym użytkowniku i ma pełnić funkcję komputera domowego, to jednak nawiązuje on do linii IBM-PC. Nie chodzi bynajmniej o niewolniczą kopię wyrobu „Big Blue". Część elektroniczna jest funkcjonalnym odpowiednikiem elementów występujących w IBM-PC, ale zrealizowano ją w inny sposób. Zamiast płyty głównej i kilku dodatkowych pakietów wetkniętych w gniazda, zrealizowanych w oparciu o typowe układy logiczne, Schneider-PC ma tylko jedną płytę montażową, jednoczącą wszystkie podstawowe funkcje mikrokomputera. Oszczędności miejsca osiągnięto dzięki trzem specjalizowanym układom logicznym LSI typu matrycy logicznej (*gate array*).

Sercem mikrokomputera jest procesor Intel-8086 (nie 8088), pracujący z częstotliwością zegarową 8 MHz, istnieje możliwość wstawienia koprocesora arytmetycznego Intel-8087. Pamięć RAM liczy 512 KB, ale na płycie głównej przewidziano podstawki, pozwalające rozbudować ją do 1 MB. Możliwości graficzne odpowiadają standardowemu pakietowi grafiki barwnej (CGA) i obejmują tryby tekstowe barwne i monochromatyczne z 40 lub 80 znakami w wierszu, tryb grafiki wysokiej rozdzielczości 640 × 200 punktów i grafiki barwnej 320 × 200 punktów. Na płycie głównej zainstalowane są interfejsy: równoległy Centronics i szeregowy RS-232. Dla ewentualnej rozbudowy systemu zarezerwowano trzy typowe gniazda magistrali, pozwalające wstawić standardowe pakiety dla IBM-PC. W wariancie najprostszym pamięć masową stanowi pojedyncza stacja dysków 5.25 cala o pojemności 360 KB — w pełni zgodna ze standardem IBM-PC. W wersjach bardziej rozbudowanych występują dwie stacje, albo stacja dysków elastycznych i dysk twardy typu Winchester o pojemności 10 lub 20 MB. Podłączenie dysku twardego wymaga jednak zainstalowania pakietu kontrolera.

Wraz z komputerem oferowany jest bogaty pakiet oprogramowania, obejmujący system operacyjny MS-DOS V3.2, GEM (*Start-Up* + *Desk-Top*), Locomotive BASIC oraz drugi system operacyjny DOS-Plus, lansowany przez Digital Research i realizujący funkcje MS-DOS, ale oferujący dodatkowe możliwości, w tym wielozadaniowość.

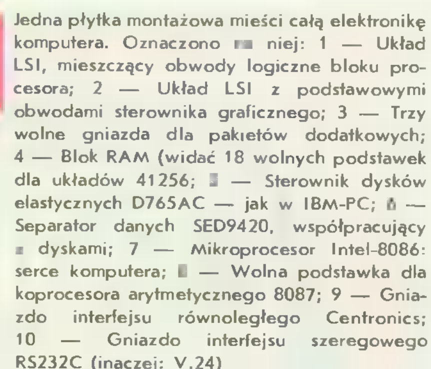
PC 1512 z monitorem barwnym i dwiema stacjami dysków elastycznych w całej okazałości



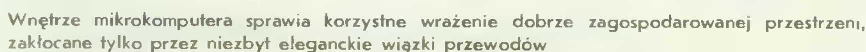
Klawiatura jest — niestety — wzorowana niewolniczo na IBM-PC i ma podobne niedostatki. W skład konfiguracji podstawowej wchodzi jednak również myszka, praktycznie niezbędna do pracy z GEM. Nabywca może wybrać monitor barwny lub monochromatyczny. Niestety, trzeba się zdecydować na monitor marki Schneider, gdyż w jego wnętrzu mieści się zasilacz komputera. Trudno to uznać za zaletę. Zrezygnowano z wentylatora, co zlikwidowało wprawdzie hałas i zredukowało cenę, ale zmniejszyło też wydajność zasilacza i znacznie pogorszyło warunki chłodzenia ewentualnie zainstalowanych pakietów dodatkowych. Dołączenie dodatkowego monitora innej produkcji, np. monitora barwnego do systemu wyposażonego w ekran monochromatyczny, napotyka trudności.

Procesor 8086 ma magistralę szesnastobitową i zapewnia prędkość przetwarzania o ok. 40% większą niż 8088 przy podobnej częstotliwości zegara. Uwzględniając fakt, że zegar pracuje 1,67 raza szybciej niż w oryginalnym IBM-PC, Schneider-PC charakteryzuje się ponad dwukrotnie większą wydajnością. Z drugiej strony mikroprocesor 8086, aczkolwiek w pełni zgodny programowo z 8088, cechuje się innymi czasami wykonywania instrukcji, występujących w różnym kontekście. Wskutek tego oraz z powodu innej częstotliwości zegara nie będzie poprawnie funkcjonować część programów dla IBM-PC, wymagających szczególnie starannego odmierzania czasu, jak np. znany i ceniony program kopiujący *Copy-Write*. Aczkolwiek Schneider-PC może korzystać ze znacznej większości oprogramowania dla IBM-PC, raczej nie należy oczekiwać 100% zgodności programowej.

Czym jest Schneider-PC? Komputerem domowym o cechach sprzętu profesjonalnego czy tanim urządzeniem profesjonalnym? Chyba jednak tym pierwszym, chociaż granica jest zatarta. Jest faktem, że w miejscu pracy solidna obudowa metalowa góruje nad plastikową, nie mówiąc o lepszym ekranowaniu i wynikającym z niego niższym poziomom zakłóceń radiowych. Możliwości rozbudowy są trochę ograniczone — brak chociażby miejsca dla równoczesnego zainstalowania dysku twardego i dwóch stacji dysków elastycznych — kompozycji dość typowej w warunkach profesjonalnych. Trzy gniazda dla dodatkowych pakietów też nie zawsze wystarczą, tym bardziej że jedno z nich zapewne trzeba będzie poświęcić dla zegara z zasilaniem bateryjnym — wyposażenia raczej niezbędnego przy intensywnej pracy. Oprócz tego bardziej „prądożer-



Jedna płytka montażowa mieści całą elektronikę komputera. Oznaczono na niej: 1 — Układ LSI, mieszczący obwody logiczne bloku procesora; 2 — Układ LSI z podstawowymi obwodami sterownika graficznego; 3 — Trzy wolne gniazda dla pakietów dodatkowych; 4 — Blok RAM (widać 18 wolnych podstawek dla układów 41256; 5 — Sterownik dysków elastycznych D765AC — jak w IBM-PC; 6 — Separator danych SED9420, współpracujący z dyskami; 7 — Mikroprocesor Intel-8086: serce komputera; 8 — Wolna podstawka dla koprocesora arytmetycznego 8087; 9 — Gniazdo interfejsu równoległego Centronics; 10 — Gniazdo interfejsu szeregowego RS232C (inaczej: V.24)



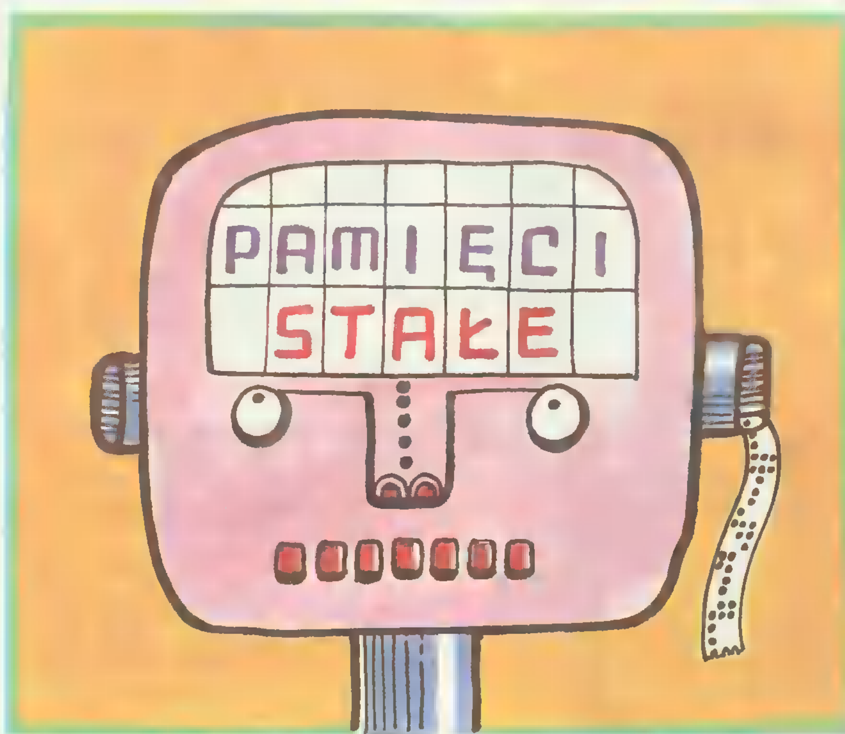
Wnętrze mikrokomputera sprawia korzystne wrażenie dobrze zagospodarowanej przestrzeni, zakłócane tylko przez niezbyt eleganckie wiązki przewodów

nym pakietom dodatkowym prędko zrobi się gorąco we wnętrzu pozbawionej wentylatora obudowy, zwłaszcza że zlokalizowano je w miejscu pozbawionym otworów wentylacyjnych.

Fakt zainstalowania sterownika graficznego na stałe będzie niestety poważnym ograniczeniem, gdy zechcemy zainstalować inny pakiet graficzny, np. monochromatyczną kartę wysokiej rozdzielczości Hercules w pełnej konfiguracji lub kartę EGA. Standardowy monitor i tak zresztą nie funkcjonowałby z tymi pakietami.

Czy warto kupić Schneidera-PC? Jeśli cenimy głównie komfort i elegancję — z pewnością tak. Mikrokomputer ten w najtańszej wersji, z monitorem monochromatycznym i jedną stacją dysków, kosztuje w RFN nieco poniżej 2000 DM. Trzeba jednak podkreślić, że dla polskiego, prywatnego nabywcy Schneider-PC jest mniej atrakcyjny niż dla zachodniego. Oprogramowanie jest zbędnym wydatkiem, gdyż można je bez większego trudu uzyskać w kraju. Specjalizowane układy scalone mogą dostarczyć poważnych problemów w razie awarii, podczas gdy typowe elementy do IBM-PC są w kraju łatwo osiągalne. Można sądzić, że specjalizowane układy LSI „potanią" produkcję masową. Zgoda — lecz Schneider-PC jest produkowany w Europie. Ten fakt stawia go na z góry przegranej pozycji w konkurencji cenowej z wyrobami *made in* Tajwan lub Korea Południowa.

Moja osobista sympatia pozostaje w każdym razie po stronie wyrobów południowoazjatyckich. Obok niższej ceny i dostępności elementów argumentem są: możliwość swobodnego wyboru konfiguracji i nieustannej jej rozbudowy, i przebudowy zależnie od potrzeb, oraz osiągalność 100% kompatybilności z IBM-PC. Łatwość otwierania blaszanej obudowy, wyposażonej w odchylaną pokrywę zamykaną na zatrzask, jest wielką zaletą przy wszelkich pracach konstruktorskich (Schneider-PC ma obudowę skręcaną wkrętami). Tym niemniej Schneider-PC zasługuje na szczególną uwagę, będąc prekursorem komputerów domowych wyposażonych we właściwości standardowego sprzętu profesjonalnego, a nawet przewyższających go niektórymi parametrami, jak np. szybkość przetwarzania. **(rw)**



Pamięci stałe

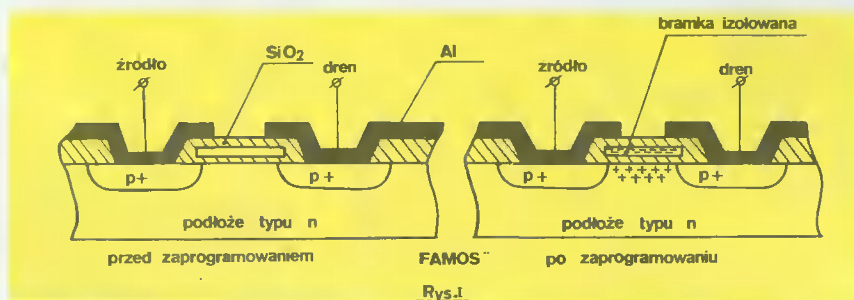
Krzysztof Wiśniewski

Każdy system mikrokomputerowy musi posiadać choć minimalną pamięć stałą, potrzebną do poprawnego rozruchu urządzenia. Pamięć ta, po załadowaniu właściwego systemu operacyjnego lub języka programowania, jest wyłączana i do dyspozycji użytkownika pozostaje cały obszar RAM-u. W pionierskim okresie prac nad komputerami zdarzało się niektórym konstruktorom systemów zapominać o tym, co wywoływało pewne emocje przy pierwszym rozruchu. W typowych komputerach domowych, nie posiadających stacji dysków elastycznych jako wyposażenia standardowego, pamięć stała zawiera rezydujący interpreter języka (zwykle BASIC), mniej lub więcej rozwinięty edytor ekranowy, procedury wejścia/ /wyjścia, (obsługa klawiatury, monitora TV, magnetofonu, drukarki) i startu.

Pamięci stałe dzielą się na kilka podstawowych typów: ROM (*Read Only Memory*), PROM (*Programmable Read Only Memory*), EEPROM (*Electrically Erasable Programmable Read Only Memory*), EPROM (*Erasable Programmable Read Only Memory*).

ROM to pamięci stałe programowane w procesie produkcyjnym przez wytwórcę na podstawie dokumentacji dostarczonej przez zamawiającego. Dla każdego typu jest konieczne wytworzenie nowych masek niezbędnych w procesie fotolitografii, dlatego produkcja staje się opłacalna przy tysiącach sztuk, a cena jednostkowa spada bardzo nisko przy seriach rzędu setek tysięcy. Przykładem są generatory znaków alfanumerycznych i pamięci stałe popularnych mikrokomputerów.

PROM są to programowane pamięci stałe wytwarzane masowo i stosowane w urządzeniach profesjonalnych, gdzie nie opłaca się stosowanie pamięci ROM programowanych maską lub istnieje potrzeba indywidualnego oprogramowania kolejnych egzemplarzy urządzenia. Programowanie ich polega na przepalaniu impulsami elektrycznymi połączeń w strukturze pamięci.



Rys.1

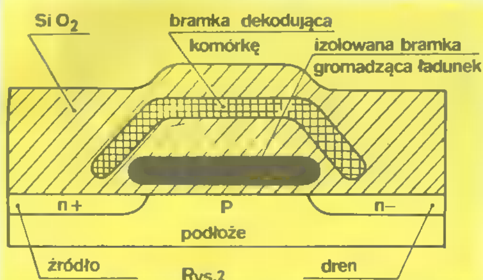
EEPROM to elektrycznie programowane i wymazywane pamięci stałe, które mimo swoich niewątpliwych zalet są mało popularne. Jest to idealna pamięć do systemów rozruchowych. Kasowanie i programowanie wymaga podobnych impulsów elektrycznych. Niestety produkcja tego typu pamięci nie jest rozwijana na tyle, aby mogły one konkurować z pamięciami EPROM.

EPROM są to pamięci ROM najbardziej popularne wśród użytkowników mikrokomputerów. Ze względu na zakres pojemności, łatwość programowania i wymazywania danych oraz niską cenę. Ich programowanie polega na podaniu impulsów elektrycznych na wejście programujące, które modyfikują MOS (*Metal Oxid Semiconductor*), w której to technologii jest zbudowana elementarna komórka pamięci. Kasowanie zapisu polega na naświetlaniu struktury pamięci promieniami ultrafioletowymi.

Prace nad pamięcią reprogramowalną rozpoczęto pod koniec lat sześćdziesiątych i dały pierwsze rezultaty w początku lat osiemdziesiątych, kiedy to firma INTEL w swoich materiałach opublikowała dane dotyczące nowego typu komórki pamięci nazwanej „FAMOS". Wkrótce w sprzedaży ukazały się pierwsze pamięci stałe reprogramowalne EPROM typu Intel 1702 o pojemności 256 bajtów. Komórka typu „Famos" jest to tranzystor MOS z kanałem typu p i izolowaną bramką. Warstwą izolacyjną o bardzo dobrych parametrach jest dwutlenek krzemu SiO_2, który jest łatwy do wytworzenia w procesie produkcyjnym. Przewodzenie między drenem a źródłem (rys. 1) w komórce nie zapisanej następuje pod wpływem powielania lawinowego. Bramka zostaje w wyniku akcji lawinowej naładowana ujemnie nosnikami elektronowymi generowanymi w złączu drenowym. Nośniki do obszaru bramki przedostają się przez kilkudziesięcionanometrowy obszar izolatora podbramkowego. Ładunek przestrzenny zgromadzony w obszarze bramki indukuje kanał przewodzący typu p pomiędzy źródłem a drenem. Jeśli teraz odłączymy od komórki napięcie powodujące przepływ nośnikow w wyniku powielania lawinowego kanał przewodzący nie zniknie dzięki „zamrożeniu" w obszarze bramki ładunku przestrzennego, który będzie podtrzymywał jego istnienie.

Komórki typu „Famos", ze względu na dużą powierzchnię, wysokie napięcie programowania i stosunkowo szybką utratę zawartości nie doczekały się wielkiej popularności. Następne generacje pamięci EPROM były wykonywane jak przedstawiono na rys. 2. Taka topologia pozwoliła na oszczędność krzemu i zwiększenie upakowania (2704, 2708), a co za tym idzie spadek ceny i zwiększoną popularność. Jednak i one nie były pozbawione pewnych wad odziedziczonych po swoich poprzedniczkach.

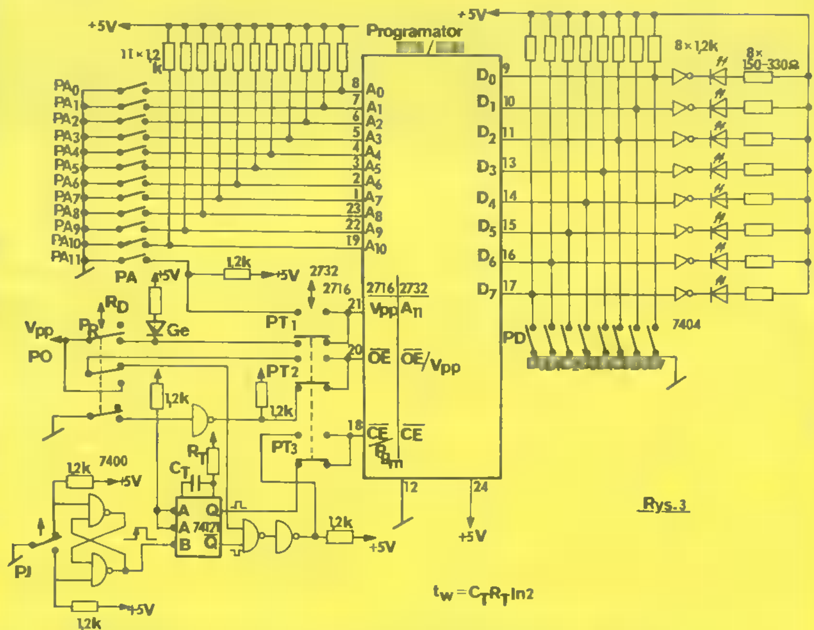
Prawidłowe zaprogramowanie komórki pamięci wymaga zgromadzenia w obsza-



Rys.2

[illegible] bramki odpowiedniego ładunku. Wiąże się z tym odpowiednio długi czas przepływu prądu lawinowego między źródłem i drenem. Może to doprowadzić do łatwego uszkodzenia struktury pamięci. Zanim nie udoskonalono budowy pamięci EPROM ładunek trzeba było wprowadzać porcjami przez czas nie dłuższy niż 1 ms, kolejno do wszystkich komórek powtarzając tą operację co najmniej 100 razy. Dopiero następne generacje pamięci poczynając od 2716 pozwalały zapisać dane jednym impulsem trwającym 50 ms. Dalszy rozwój doprowadził do powstania pamięci w oparciu o technologię HMOS (Intel) i CMOS. Zwiększono szybkość działania i czas dostępu do danych zmniejszył się z 1 μs do 200 ns, [illegible] w niektórych specjalnych rozwiązaniach oscyluje w pobliżu 100 ns. Napięcia programujące zmniejszały się od 25 V poprzez 21,5 V do 12,5 V.

Trwałość zapisu zwiększyła się wielokrotnie, jednak na razie jest ona trudna do oceny ze względu na zbyt krótki czas korzystania z tych elementów. Nowsze typy pamięci EPROM (powyżej 2732) pozwalają na programowanie przy zwiększonym napięciu zasilania do 6 V i napięciu programującym 12,5 V.



Rys.3

Jeśli chcemy zaprogramować małą pamięć EPROM potrzebną do przetestowania budowanego mikrokomputera lub programu dla pozytywki, wystarczy zbudować prosty ręczny programator (ang.: *only manual promer*). Zaprogramowanie

[illegible] stat. CMOS																					
			EPROM typu 27 XXX																		
62256	6264	6216	27512	27256	27128	2764	2732	2716	2708			2708	2716	2732	2764	27128	27256	2[illegible]	6216	6264	62256
A_{14}	NC		A_{15}	V_{pp}	V_{pp}	V_{pp}				1	28				+5V	+5V	+5V	+5V		+5V	+5V
A_{12}	A_{12}		A_{12}	A_{12}	A_{12}	A_{12}				2	27				$\overline{PGM}$	$\overline{PGM}$	A_{14}	A_{14}		$\overline{WE}$	$\overline{WE}$
A_7	A_7	A_7	A_7	A_7	A_7	A_7	A_7	A_7	A_7	3	26	+5V	+5V	+5V	N.C	A_{13}	A_{13}	A_{13}	+5V	CE	A_{13}
A_6	A_6	A_6	A_6	A_6	A_6	A_6	A_6	A_6	A_6	4	25	A_8	A_8	A_8	A_8	A_8	A_8	A_8	A_8	A_8	A_8
A_5	A_5	A_5	A_5	A_5	A_5	A_5	A_5	A_5	A_5	5	24	A_9	A_9	A_9	A_9	A_9	A_9	A_9	A_9	A_9	A_9
A_4	A_4	A_4	A_4	A_4	A_4	A_4	A_4	A_4	A_4	6	23	–5V	Vpp	A_{11}	A_{11}	A_{11}	A_{11}	A_{11}	$\overline{WE}$	A_{11}	A_{11}
A_3	A_3	A_3	A_3	A_3	A_3	A_3	A_3	A_3	A_3	7	22	$\overline{CS}/WE$	$\overline{OE}$	$\overline{OE}/V_{pp}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}/V_{pp}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$
A_2	A_2	A_2	A_2	A_2	A_2	A_2	A_2	A_2	A_2	8	21	+12V	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}
A_1	A_1	A_1	A_1	A_1	A_1	A_1	A_1	A_1	A_1	9	20	PRG	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
A_0	A_0	A_0	A_0	A_0	A_0	A_0	A_0	A_0	A_0	10	19	D_7	D_7	D_7	D_7	D_7	D_7	D_7	D_7	D_7	D_7
D_0	D_0	D_0	D_0	D_0	D_0	D_0	D_0	D_0	D_0	11	18	D_6	D_6	D_6	D_6	D_6	D_6	D_6	D_6	D_6	D_6
D_1	D_1	D_1	D_1	D_1	D_1	D_1	D_1	D_1	D_1	12	17	D_5	D_5	D_5	D_5	D_5	D_5	D_5	D_5	D_5	D_5
D_2	D_2	D_2	D_2	D_2	D_2	D_2	D_2	D_2	D_2	13	16	D_4	D_4	D_4	D_4	D_4	D_4	D_4	D_4	D_4	D_4
GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	14	15	D_3	D_3	D_3	D_3	D_3	D_3	D_3	D_3	D_3	D_3

Rys. 4

podczas jednego posiedzenia całej pamięci 2 kB, a tym bardziej 4 kB jest rzeczą trudną jeśli nie chcemy zrobić tego byle jak, ale przy odrobinie cierpliwości i kilku podejściach pewnie się to uda. Od czegoś trzeba zacząć i nie każdy jest „profesjonalistą" na koszt swojej szkoły lub zakładu pracy. Układ można w prosty sposób zmodyfikować w zależności od inwencji i posiadanego zapasu części. Można go przystosować do programowania pamięci PROM typu 74188 itp.... Programator wyposażony jest w trzy grupy przełączników.

Przełącznik zadający adres komórki pamięci, do której chcemy zapisać daną lub ją odczytać (można go zastąpić licznikiem rewersyjnym z generatorem do szybkiej zmiany jego zawartości i przyciskiem generującym impuls jednorazowy).

Przełącznik zadający wartość danej połączony z wyświetlaniem. Przełączniki ustalające typ pamięci EPROM, wyzwalające impuls zapisu zmieniające tryb pracy z zapisu na odczyt. Czwartą ważną częścią programatora jest multiwibrator monostabilny generujący impuls programujący o długości 50 ms. Pamiętać należy o nieprzekraczaniu zastrzeżonego przez producenta czasu programowania wynoszącego 55 ms. W wielu wypadkach wystarczy czas dużo krótszy, ale przy ręcznym programowaniu jest to nieistotna oszczędność, natomiast przedłużenie może trwale uszkodzić pamięć na skutek przekroczenia mocy strat cieplnych. Opisywany programator może programować pamięci serii Intel 2716 lub 2732, 2732 A. Pamiętać należy o odpowiednim napięciu V_{pp} w zależności od typu programowanej pamięci. Na rys. 3 przedstawiony jest schemat programatora. Zamiast odczytu bezpośredniego binarnego można zastosować dwie pamięci PROM jako dekodery.

Można to samo zadanie wykonać za pomocą standardowych układów TTL. (Zostało to opisane w książce: J. Pieńkos J. Turczyński *„Układy scalone TTL w systemach cyfrowych"*).

Programowanie zaczynamy od podłączenia odpowiedniego napięcia programującego, ustawienia adresu i danych. Następnie wyzwalamy impuls programujący. Chcąc sprawdzić zawartość zapisanej komórki pamięci zmieniamy pozycję przełącznika PO na odczyt.

Programator może również służyć do odczytów fabrycznie zapisanych pamięci z różnego rodzaju przystawek i interfejsów. Chcąc kopiować pamięci ROM lub EPROM wystarczy dołączyć drugą podstawkę z przełącznikiem adresów, wysterować odpowiednimi sygnałami końcówki OE i CE oraz połączyć bezpośrednio wyprowadzenia odpowiednich linii danych.

Dalsza rozbudowa mija się z celem. Należy dążyć do zatrudnienia w procesie programowania posłusznego i cierpliwego robotnika jakim jest mikrokomputer, któremu nie będzie się nudziło wykonywanie kilku tysięcy tych samych czynności, a na dodatek zajmie mu to tylko kilka minut.

Na rys. 4 przedstawiamy zestawienie topologii najbardziej popularnych pamięci EPROM i statycznych pamięci RAM.

CIEKAWE KSIĄŻKI

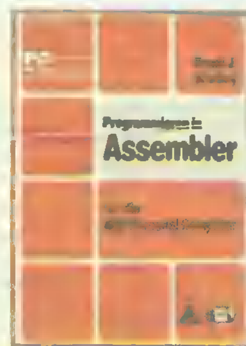
Bohdan Frelek, Andrzej Lewandowski, MIKROKOMPUTER — PROGRAMOWANIE W JĘZYKU BASIC, SIGMA 1986. Wydana w nakładzie 60 tys. egzemplarzy pozycja jest adresowana do szerokiego kręgu odbiorców, pragnących zgłębić podstawy programowania w BASIC-u. Można ją polecić nawet zupełnie początkującym. Szczególnie usatysfakcjonowani będą jednak zapewne użytkownicy Commodore C-64, gdyż autorzy oparli się o dialekt BASIC-a, stosowany właśnie w tym mikrokomputerze. Dialektom ZX-81, ZX-Spectrum i MERITUM poświęcone są oddzielne podrozdziały.

Po krótkim wprowadzeniu w zasady funkcjonowania mikrokomputera i objaśnieniu podstawowych pojęć, Autorzy przechodzą do prostych obliczeń w trybie kalkulatorowym, oswajając Czytelnika w ten sposób z maszyną. Następne rozdziały omawiają podstawowe rozkazy i ich zastosowanie, metody tworzenia programów, ich wprowadzania i uruchamiania, operacje na tekstach oraz operacje wejścia wyjścia. Rozdział siódmy, poświęcony grafice i efektom dźwiękowym, odnosi się wyłącznie do C-64 i jest nasycony rozkazami PEEK i POKE. Autorzy używają też SIMON'S BASIC-a. Rozdział ostatni jest dedykowany bardziej zaawansowanym Czytelnikom, poruszając takie tematy jak wewnętrzna struktura programów i techniki optymalizacyjne.

Książka została napisana w sposób bardzo przystępny i komunikatywny, zastosowanie poszczególnych instrukcji i zleceń zostało odpowiednio zilustrowane przykładami. Można ją więc polecić wszystkim zainteresowanym. Są też niestety i uwagi krytyczne. Podstawowy zarzut dotyczy znikomej dbałości autora o styl programowania, małej troski o właściwą strukturę i czytelność programów. Autorzy zachęcają natomiast do zagęszczania programów przez eliminację odstępów i komasację instrukcji w jednym wierszu, polecają wielokrotne wykorzystywanie zmiennych do różnych celów, itd. Wszystko to ma prowadzić do zaoszczędzenia cennych komórek pamięci. Czy BASIC jest jednak narzędziem do tworzenia dużych programów? Jeżeli nie, to takie bałamutne wskazówki wpoją w niedoświadczonego Czytelnika przekonanie, ża najlepszy programista to taki, który pisze najkrótsze programy. Jest jednak nadzieja, że bardziej krytyczni Czytelnicy zrezygnują ze wspomnianej „optymalizacji", odstraszeni wyglądem niektórych listingów, zamieszczonych w książce. (rw)

David J. Bradley PROGRAMMIEREN IN ASSEMBLER FUER DIE IBM PERSONAL COMPUTER, Carl Hanser Verlag, Monachium, RFN, 1986 r.

David J. Bradley był członkiem zespołu, który skonstruował IBM-PC. Trudno sobie wyobrazić bardziej kompetentnego autora pozycji o programowaniu IBM-PC XT w języku asemblera. Jego wydana w języku niemieckim książka jest adresowana do kręgu osób, które zdobyły już pewne obycie w kontaktach z komputerem, posiadły podstawowy aparat pojęciowy i terminologię.



Po krótkim przypomnieniu podstawowych pojęć i podstaw techniki komputerowej (system dwójkowy i szesnastkowy, arytmetyka dwójkowa, stos itd.), Autor przystępuje do prezentacji mikroprocesora Intel-8088. Następnie w systematyczny sposób przedstawiono listę rozkazów. Poszczególne grupy rozkazów ilustrowane są listingami, przedstawiającymi zarazem sposób ich kodowania. Przy okazji wyjaśnione są konwencje notacji asemblerowej.

Następne rozdziały są poświęcone praktycznemu posługiwaniu się asemblerem. Po zwięzłym omówieniu systemu PC-DOS, przerwań systemowych, formatów zbiorów, edytora EDLIN i debuggera DEBUG autor przedstawia metody korzystania z mechanizmów makroasemblera, w tym: makroinstrukcji i struktur danych.

Koprocesor arytmetyczny 8087 zaprezentowano w odrębnym, obszernym rozdziale, objaśniając jego strukturę wewnętrzną, formaty wykładnicze i listę rozkazów. Podstawowe techniki programowania 8087, w tym konwersja liczb wykładniczych na postać znakową i uruchamianie programów, zilustrowano listingami.

Pozostała część książki poświęcona jest już zdecydowanie IBM-PC XT. Pokrótce przedstawiono konfigurację sprzętu, funkcjonowanie i zasady programowania poszczególnych układów peryferyjnych, w tym kontrolera ekranu 6845, interfejsów: szeregowego i równoległego oraz DMA i kontrolera dysków elastycznych. W oddzielnym rozdziale omówiono zawartość ROM-BIOS, koncentrując się na jego istotnych funkcjach i podstawowych wywołaniach. Ostatni rozdział referuje metody rozszerzania funkcji BIOS oraz łączenia procedur asemblerowych z BASIC-em i innymi językami programowania. Dodatki zawierają pełną listę rozkazów procesorów 8088 i 8087.

Książka jest nie tylko bardzo wartościowa merytorycznie, ale i komunikatywna. Można z niej wiele skorzystać nie mając dostępu do komputera. Krótko: pozycja godna polecenia wszystkim użytkownikom IBM-PC XT władającym językiem niemieckim. (rw)

KOMPUTEROWE „PRZYPOMINADŁO"

Tadeusz Basista

Program ten jest przeznaczony dla użytkowników mikrokomputera ZX Spectrum 48k, którzy opanowali umiejętność programowania w języku BASIC w stopniu umożliwiającym tworzenie własnych programów użytkowych i którym przestały wystarczać możliwości wbudowanego interpretera. Najczęstszym tego powodem jest brak specjalnych instrukcji umożliwiających efektywne wykorzystanie mikrokomputera lub zbyt wolna realizacja istniejących instrukcji.

Radykalnym rozwiązaniem tego problemu jest pisanie programu w asemblerze lub językach wyższego rzędu, np. w Pascalu, jednak ze względów ekonomicznych w wielu typowych przypadkach nie jest to uzasadnione. Jednym ze sprawdzonych rozwiązań jest łączenie programu głównego napisanego w BASIC-u (realizowanego przez interpreter) z procedurami napisanymi w asemblerze (realizowanymi przez procesor).

W każdym większym programie można wyodrębnić 5—10 procent kodu „pożerającego" nawet 90 procent czasu działania programu. Przykładem może być problem sortowania pliku lub odnajdywania rekordu w programach przetwarzania danych. W programach dokonujących obliczeń numerycznych typowym fragmentem są powtarzające się pętle, wewnątrz których są wykonywane standardowe operacje arytmetyczne. W programach edukacyjnych słabym punktem może okazać się grafika, która tutaj w dużym stopniu stanowi o atrakcyjności programu.

Artykuł ten otwiera cykl problemowy przeznaczony również dla Czytelników kursu asemblera. Z tego powodu wszystkie programy maszynowe będą przedstawione przede wszystkim w postaci źródłowej i miarę możliwości szczegółowo komentowane. Posiadanie wersji źródłowej umożliwi założenie biblioteki procedur, które można będzie dołączać do innych programów.

Zastosowanie procedury

Procedura opisana poniżej umożliwia wykonywanie szybkich operacji na obrazie w trakcie działania programu głównego. Może znaleźć zastosowanie np. w programach głównych do szybkiego wyprowadzania komunikatów lub rysunków na ekran. Typowym przykładem jest tzw. strona pomocy (ang.: *help*) „ściągawka" przedstawiająca menu programu, pojawiająca się na ekranie po naciśnięciu odpowiedniego klawisza. Ze względu na uniwersalność zastosowań wykorzystano tu kombinacje klawiszy SYMBOL SHIFT +SPACE (taka kombinacja klawiszy nie jest przez interpreter używana). Ponadto nasza „ściągawka" jest widoczna na ekranie tak długo, jak oba klawisze są wciśnięte. Po ich zwolnieniu obraz wraca do stanu poprzedniego. W opisywanej procedurze przewidziano również możliwość wykorzystania fragmentów obrazu od zadanego wiersza początkowego do zadanego końcowego (0—23) co symuluje tzw. okna. Procedura może być także wywoływana z programu głównego bez konieczności użycia klawiatury. Kilka przykładowych zastosowań procedury przedstawiono w programie demonstracyjnym.

Ładowanie kodu procedury

Ponieważ część Czytelników może nie mieć jeszcze dostępu do asemblera „Gens 3", przygotowano również program ładujący kod procedury z interpretera. Dla tych, którzy nie będą wnikać w szczegóły techniczne programowania w asemblerze, podaję kilka niezbędnych informacji.

Kod procedury jest wpisywany od adresu 64908, natomiast RAMTOP jest ustawiony na 64619 — obszar od adresu 64620 do 64907 jest obszarem roboczym procedury. Obszar ten o długości 288 bajtów jest używany w charakterze bufora celem przyspieszenia operacji zamiany obrazów (256 bajtów na obraz jednego wiersza ekranu + 32 bajty na atrybuty tego wiersza). Obszar od adresu 65350 do 65535 nie jest przez procedurę wykorzystywany, można więc używać znaków graficznych zdefiniowanych znajdujących się pod standardowymi adresami.

```
    REM [illegible]
    REM [illegible] 198[illegible]
 [illegible]
 10 CLEAR [illegible] ADR=[illegible] LET SUMA=[illegible]
    FOR I=1 TO [illegible]
      READ D$
      FOR J=1 TO 31 STEP [illegible]
        LET D1=CODE D$(J)
        LET D2=CODE D$(J+1)
        LET D1=D1-48-7*(D1[illegible]
        LET D2=D2-48-7*(D2>64)
        LET D=16*D1+D2
100     POKE ADR,D
110     LET ADR=ADR+1
120     LET SUMA=SUMA+D
      NEXT J
1[illegible] NEXT I
    IF SUMA<>57761 THEN PRINT "BLAD !": STOP
    SAVE "OBRAZY M/C"CODE 6490[illegible],443
 [illegible]
    DATA "[illegible]F5C5D5E5DDE5FDE5CD[illegible]CFF2015CD5[illegible]"
    DATA "FE[illegible]D3[illegible]CFF[illegible]F8CD52FEFDE1DDE1E1D1C1"
    DATA "F1FB[illegible]FDE1DDE1E1D1C1F1FFFBC9[illegible]A5[illegible]"
    DATA "5C[illegible]323E55E[illegible]356D5CD33FEE109EBE173"
    DATA "[illegible]722AF4FDE5[illegible]2F4FDE1E5C5CD5516"
    DATA "[illegible]1E11E2A7[illegible]32[illegible]0E78B120F9C9181A0000"
[illegible]60 DATA "00150001000100000000[illegible]EFDED47ED5E"
[illegible]70 DATA "C[illegible]CFD[illegible]CFDED56C93AEAFDFE002811FE"
[illegible]80 DATA "012[illegible]F2FE0228E3FE0[illegible]283BFE042[illegible]9FC9"
[illegible]90 DATA "[illegible]A5[illegible]5C2323E55E2356E1E51922F4FDE1"
[illegible]00 DATA "[illegible]32[illegible]22F2FDC93AEBFD4721000AFB8"
[illegible]10 DATA "[illegible]3061120011910FDE5C1C9CD33FE2AF2"
[illegible]20 DATA "FD0922F0FDC921EEFD3600CD47FECD61"
[illegible]30 DATA "FECD47FEC921EEFD3AEDFDBE38563AEC"
[illegible]40 DATA "FDBE3804280218460D216CFC3E0132EF"
[illegible]50 DATA "FDCDC1FECDF3FE210058CDE3FECD1EFF"
```

```
360 DATA "DD2AF0FD3E02[illegible]2EFFD[illegible]C1FECDF3FE[illegible]1"
370 DATA "0058CDE3FECD1EFFED5BF0FD216CFC01"
380 DATA "[illegible]001EDB0[illegible]AF0FD11200119[illegible]F0FC[illegible]1EE"
390 DATA "FD3418A4C93AEEFDFE1028143012[illegible]E08"
400 DATA "[illegible]8073005 100401811D60[illegible]10040180A"
410 DATA "D640[illegible]100501803[illegible]AEEFD11200[illegible]C4,05"
420 DATA "[illegible]3031918FAEBC[illegible]06FF047[illegible]FE08[illegible]E"
430 DATA "FF0C79FE2028F2C5E1193AEFFDFE0228"
440 DATA "087EDD770[illegible]DD[illegible]18E8DD7E0077DD[illegible]18"
450 DATA "E0C9EB06[illegible]03AEFFDFE02280A7EDD7700"
460 DATA "DD[illegible]2[illegible]10F0C9DD7E0077DD2[illegible]10[illegible]C"
470 DATA "CD8E027AFE18C07BFE20C9000000000000"
```

A teraz kilka uwag dla wpisujących program źródłowy w asemblerze „Gens 3". Po załadowaniu i uruchomieniu asemblera zgodnie z przedstawionym w „MT-InforMik" opisem edytora wprowadzamy kolejne linie programu źródłowego. Można ewentualnie opuścić wszelkie komentarze (teksty po znaku średnika). Następnie dokonujemy asemblacji programu przy pomocy zlecenia „A", podając po pytaniu: „Table size?" — 1000 i po pytaniu: „Options?" — 1. Jeśli asembler nie wykrył błędow, to przede wszystkim zapisujemy na taśmę program źródłowy przy pomocy zlecenia „P", a następnie po powrocie do „Basica" (po wykonaniu zlecenia „B") zapisujemy na taśmę wygenerowany kod maszynowy.

```
10 ;PROGRAM ZAMIANY OBRAZOW
20 ; Sinclair ZX SPECTRUM
30 ;(C)Tadeusz Basista 1986
40 ; Start programu 65000
50                ORG  64620
60 PROG           EQU  23635
70 KEYSCA         EQU  #028E
80 MKROOM         EQU  #1655
90 BUFOR          DEFS 288
100 ;
110 ;Obsluga przerwania
120 $PRZER
130               DI
140               PUSH AF
150               PUSH BC
160               PUSH DE
170               PUSH HL
180               PUSH IX
190               PUSH IY
200               CALL $KLAW
210               JR   NZ,PRZER3
220               CALL $ZMIAN
230 PRZER1
240               CALL $KLAW
250               JR   Z,PRZER1
260               CALL $ZMIAN
270 PRZER2
280               POP  IY
290               POP  IX
300               POP  HL
310               POP  DE
320               POP  BC
330               POP  AF
340               EI
350               RET
360 PRZER3
370               POP  IY
380               POP  IX
390               POP  HL
400               POP  DE
410               POP  BC
420               POP  AF
430               RST  #38
440               EI
450               RET
460
470  Rezerwacji pamieci
480 $REZER
490               LD   HL,(PROG)
500               INC  HL
510               INC  HL
520               PUSH HL
530               LD   E,(HL)
540               INC  HL
550               LD   D,(HL)
560               PUSH DE
570
```

```
2710              JR   ADR1
2720 ;
2730 $ADRAT       LD   A,(EKRAKT)
2740 ADR1         LD   DE,#0020
2750              INC  A
2760              LD   B,A
2770 ADR2         DEC  ■
2780              JR   Z,ADR3
2790              ADD  HL,DE
2800              JR   ADR2
2810 ADR3         EX   DE,HL
2820              RET
2830
2840  Zamiana jednego wiersza
2850
2860 $ZAMLI
2870              LD   B,#FF
2880 ZAMLI1       INC  B
2890              LD   A,B
2900              CP   8
2910              JR   Z,ZAMLI5
2920
2930              LD   C,#FF
2940 ZAMLI2       INC  C
2950              LD   A,C
2960              CP   32
2970              JR   Z,ZAMLI1
2980
2990              PUSH BC
3000              POP  HL
3010              ADD  HL,DE
3020
3030              LD   A,(WSKZAM)
3040              CP   2
3050              JR   Z,ZAMLI4
3060 ;
3070 ZAMLI3       LD   A,(HL)
3080              LD   (IX),A
3090              INC  IX
3100              JR   ZAMLI2
3110 ;
3120 ZAMLI4       LD   A,(IX)
3130              LD   (HL),A
3140              INC  IX
3150              JR   ZAMLI2
3160 ZAMLI5       RET
3170
3180
3190 $ZAMAT
3200 ;Zamiana atrybutow
3210              EX   DE,HL
3220              LD   B,32
3230 ZAMAT1
3240              LD   A,(WSKZAM)
3250              CP   2
3260              JR   Z,ZAMAT3
3270 ;
3280 ZAMAT2       LD   A,(HL)
3290              LD   (IX),A
3300              INC  IX
3310              INC  HL
3320              DJNZ ZAMAT1
3330              RET
3340
3350 ZAMAT3       LD   A,(IX)
3360              LD   (HL),A
3370              INC  IX
3380              INC  HL
3390              DJNZ ZAMAT1
3400              RET
3410 ;
3420 ;Test nacisn.klawiszy
3430 ;SYMBOL SHIFT + SPACE
3440 $KLAW
3450              CALL KEYSCA
3460              LD   A,D
3470              CP   24
3480              RET  NZ
3490              LD   A,E
3500              CP   32
3510              RET
3520
3530  *** koniec programu ***
580 ;Zmiana licz. linii zer
590               CALL $MNOZ
600               POP  HL
610               ADD  HL,BC
620               EX   DE,HL
630               POP  HL
640               LD   (HL),E
650               INC  HL
660               LD   (HL),D
670
```

```
680 ;Zmiana adr markera
690              LD   HL,(PAMKON)
700              PUSH HL
710              ADD  HL,BC
720              LD   (PAMKON),HL
730 ;
740 ;Odsun.Basic do gory
750 ; HL-odkad,BC-ile bajtow
760              POP  HL
770              PUSH HL
780              PUSH BC
790              CALL MKROOM
800 ;
810 ;Wypelnienie pamieci
820              POP  BC
830              POP  HL
840              LD   E,"*"
850 REZER1       LD   (HL),E
860              INC  HL
870              DEC  BC
880              LD   A,B
890              OR   C
900              JR   NZ,REZER1
910              RET
920 ;
930 ;****************
940 ;*START PROGRAMU*
950 ;****************
960              JR   $MENU
970 ;
980 ;Parametry zlecen
990 KODINS       DEFB 0
1000 OBRNUM      DEFB 0
1010 EKRPOC      DEFB 0
1020 EKROST      DEFB 21
1030 ;kod zlecenia
1040 ;nr wiersza obrazu w pam.
1050 ;nr pocz.wiersza ekranu
1060 ;nr konc.wiersza ekranu
1070 ;
1080 ;Zmienne robocze
1090 EKRAKT      DEFB 0
1100 WSKZAM      DEFB 1
1110 PAMAKT      DEFS 2
1120 PAMPOC      DEFS 2
1130 PAMKON      DEFS 2
1140 ;nr aktual.wiersza ekranu
1150 ;wskaz. kierunku zamiany
1160 ;adres aktual.wiersza pam
1170 ;adres pocz.pamieci obraz
1180 ;adres konc.pamieci obraz
1190 ;
1200 ;Ustaw trybu 2 przerwan
1210 $TRYB2
1220             LD   A,#FD
1230             LD   I,A
1240             IM   2
1250             RET
1260
1270             DEFS 2
1275 ;UWAGA ! Ponizsze slowo
1276 ;musi miec lokacje #FDFF
1280             DEFW $PRZER
1290 ;
1300 ;Ustaw.trybu 1 przerwan
1310 $TRYB1
1320             IM   1
1330             RET
1340 ;
1350 $MENU
1360             LD   A,(KODINS)
1370             CP   0
1380             JR   Z,$INICJ
1390             CP   1
1400             JR   Z,$TRYB1
1410             CP   2
1420             JR   Z,$TRYB2
1430             CP   3
1440             JR   Z,$ZMIAN
1450             CP   4
1460             JR   Z,$REZER
1470             RET
1480 ;
1490 ;Odczytanie poczatku i
1500 ;konca pamieci obrazow
1510 $INICJ
1520             LD   HL,(PROG)
1530             INC  HL
1540             INC  HL
1550             PUSH HL
1560             LD   E,(HL)
1570             INC  HL
1580             LD   D,(HL)
1590             POP  HL
1600             PUSH HL
1610             ADD  HL,DE
1620             LD   (PAMKON),HL
1630             POP  HL
1640             INC  HL
1650             INC  HL
1660             INC  HL
1670             LD   (PAMPOC),HL
1680             RET
1690
1700 $MNOZ
1710             LD   A,(OBRNUM)
1720             LD   B,A
1730             LD   HL,0
1740             XOR  A
1750             CP   B
1760             JR   Z,MNOZ2
1770             LD   DE,288
1780 MNOZ1       ADD  HL,DE
1790             DJNZ MNOZ1
1800 MNOZ2       PUSH HL
1810             POP  BC
1820             RET
1830 ;
1840 ;Obliczenie adresu danego
1850 ;wiersza w pamieci obrazu
1860 $USTPA
1870             CALL $MNOZ
1880             LD   HL,(PAMPOC)
1890             ADD  HL,BC
1900             LD   (PAMAKT),HL
1910             RET
1920 ;
1930 ;Zamiana ekranu z pamiecia
1940 $ZMIAN
1950             LD   HL,EKRAKT
1960             LD   (HL),0
1970             CALL $USTPA
1980             CALL $LINIA
1990             CALL $USTPA
2000             RET
2010
2020 ;Zamiana kolej. wierszy
2030 $LINIA
2040             LD   HL,EKRAKT
2050 LINIA1      LD   A,(EKROST)
2060             CP   (HL)
2070             JR   C,LINIA4
2080             LD   A,(EKRPOC)
2090             CP   (HL)
2100             JR   C,LINIA2
2110             JR   Z,LINIA2
2120             JR   LINIA3
2130 ;
2140 ;Linia ekranu do bufora
2150 LINIA2
2160             LD   IX,BUFOR
2170             LD   A,1
2180             LD   (WSKZAM),A
2190             CALL $ADRLI
2200             CALL $ZAMLI
2210             LD   HL,#5800
2220             CALL $ADRAT
2230             CALL $ZAMAT
2240 ;
2250 ;Pamiec na ekran
2260             LD   IX,(PAMAKT)
2270             LD   A,2
2280             LD   (WSKZAM),A
2290             CALL $ADRLI
2300             CALL $ZAMLI
2310             LD   HL,#5800
2320             CALL $ADRAT
2330             CALL $ZAMAT
2340 ;
2350 ;Bufor do pamieci
2360             LD   DE,(PAMAKT)
2370             LD   HL,BUFOR
2380             LD   BC,288
2390             LDIR
2400 ;
2410 ;Adres nastepnego wiersza
2420             LD   HL,(PAMAKT)
2430             LD   DE,288
2440             ADD  HL,DE
2450             LD   (PAMAKT),HL
2460 ;
2470 LINIA3      LD   HL,EKRAKT
2480             INC  (HL)
2490             JR   LINIA1
2500 LINIA4      RET
2510 ;
2520 $ADRLI
2530 ;Oblicz adresu poczatku
```

```
2540 ,wiersza na ekranie
2550            LD      A,(EKRAKT)
2560            CP      16
2570            JR      Z,CZE3
2580            JR      NC,CZE3
2590            CP      8
2600            JR      Z,CZE2
2610            JR      NC,CZE2
2620 CZE1       LD      HL,#4000
2630            JR      ADR1
2640 ,
2650 CZE2       SUB     8
2660            LD      HL,#4800
2670            JR      ADR1
2680 ;
2690 CZE3       SUB     16
2700            LD      HL,#5000
```

Wywołanie i parametry procedury

Każde wywołanie procedury realizujemy instrukcją RANDOMIZE USR R, gdzie R = 65000. Rodzaj zlecenia realizowanego przez procedurę okreslamy instrukcją POKE R+2, n (gdzie n = 0, 1, 2, 3, 4). Parametry zleceń wprowadzamy instrukcją POKE w komórki R+3, R+4, R+5. Procedura realizuje następujące zlecenia:

a) n = 0 Inicjacja procedury. Zostaje zapamiętany adres początku i końca obszaru pamięci obrazów. Obszar ten będzie rezerwowany za REM w pierwszej linii programu BASIC, nie należy więc tej linii usuwać z programu.

b) n = 4 POKE R+3,k Rezerwacja obszaru pamięci na k wierszy ekranu. W linii 1 (REM) zostanie zarezerwowanych k ⨯ 288 bajtów.

c) n = 3 POKE R+3,k POKE R+4,m POKE R+5,n. Wiersze obrazu od m do n zostaną zamienione z pamięcią obrazu od numeru rezerwacji k. Uwaga! Pierwszy wiersz w pamięci ma numer k = 0, ponadto Ø < = m < = n < = 23.

d) n = 2 Uruchomienie trybu 2 przerwań.

e) n = 1 Uruchomienie trybu 1 przerwań.

Uwagi:

1) Rezerwacji pamięci można dokonywać wielokrotnie. Kolejne obszary będą rezerwowane za już istniejącymi.
2) Wprzypadku współpracy z urządzeniami zewnętrznymi oraz przy zakończeniu programu zaleca się przywrócenie trybu 1 przerwań.
3) Zaleca się zmianę numeru 1 linii Basic zawierającej REM na numer 0. Można to osiągnąć wykonując zlecenie:
   POKE PEEK 23635+256 ⨯ PEEK 23636+1, Ø

Ilustracją prostego wykorzystania procedury jest poniższy program demonstracyjny:

```
  1 REM
  2 REM Program demonstracyjny
  3 REM (C) Tadeusz Basista
  4
 10 CLEAR 64619: LOAD "OBRAZY M/C"CODE 64908,443
 12 BORDER 7: PAPER 7: INK 0: CLS
 20 LET R=65000: RANDOMIZE USR R
 30 GO SUB 400: GO SUB 500
 40 PAUSE 300: GO SUB 600
 50 GO SUB 700: GO SUB 800
 99
100 REM Program malowania
102 BORDER 0: PAPER 0: INK 7: CLS : POKE 23658,8
104 LET W=0: GO SUB 200
110 PLOT  OVER W;X,Y
112 PRINT #1;AT 0,0; INVERSE W;"X=";X,TAB 9;"Y=";Y
120 LET I$=INKEY$: IF I$="" THEN  GO TO 120
122 IF I$="P" THEN  LET X=X+1*(X<255): GO TO 110
124 IF I$="O" THEN  LET X=X-1*(X>0): GO TO 110
126 IF I$="Q" THEN  LET Y=Y+1*(Y<175): GO TO 110
128 IF I$="A" THEN  LET Y=Y-1*(Y>0): GO TO 110
```

```
[illegible] THEN  GO TO 1[illegible]
[illegible] THEN  GO SUB [illegible] TO 110
[illegible] THEN  GO SUB [illegible] TO 110
[illegible] "J" THEN  GO SUB [illegible] GO TO 110
[illegible] "W" THEN  LET W=W+1*(W=0)-1*(W=1) [illegible] TO 11[illegible]
[illegible] THEN  GO SUB [illegible]00 [illegible]
[illegible]
[illegible] malowania
[illegible] #1 AT 1,0 [illegible]
[illegible] 0 OR X 255 THEN [illegible]
[illegible] INPUT #1,AT 1,0 "Y=",Y
[illegible] 0 OR Y 17[illegible] THEN [illegible]
[illegible] RETURN
[illegible] Zapis rysunku [illegible]
[illegible] SUB 900
[illegible] "RYSUNEK" SCREEN$
[illegible] GO SUB 900
[illegible] RETURN
[illegible]
[illegible] REM Odczyt obrazu [illegible]
[illegible] GO SUB 900
[illegible] LOAD "RYSUNEK" SCREEN$
[illegible] GO SUB 900
[illegible] RETURN
[illegible]
400 REM Rezerwacja pamieci na 5 wierszy ekranu
410 POKE R+2,4: POKE R+3,5: RANDOMIZE USR R
[illegible] RETURN
[illegible] REM Przygotowanie strony pomocy
[illegible] CLS : PRINT AT 0,0;TAB 10; FLASH 1;"*** MENU ***"
[illegible] PRINT AT 1,0;"Q-gora  W-[illegible]er   O-lewo  P-prawo"
[illegible] PRINT AT [illegible],0 "A-dol  S-save   J-load  L-start"
[illegible] PRINT AT [illegible] 0 "SS+SPACE -menu   V-cls   K-stop"
[illegible] PRINT AT 4,0;"Wspolrzedne  0<[illegible] 0<=[illegible]=175"
[illegible] RETURN
[illegible]
[illegible] REM [illegible]owanie stron pomocy
[illegible] POKE R+2,3: POKE R+3,0
[illegible] POKE R+4,0: POKE R+5,4: RANDOMIZE USR R
[illegible] RETURN
[illegible]
[illegible] REM Ustawienie param. do wydruku stron [illegible]
[illegible] POKE R+3,0: POKE R+4,10: POKE R+5,14
[illegible] RETURN
[illegible]
[illegible] REM Tryb [illegible] przerwan
[illegible] POKE R+2,[illegible]: RANDOMIZE USR [illegible]
[illegible] RETURN
[illegible]
[illegible] REM Tryb [illegible] przerwan
[illegible] POKE R+2,[illegible]: RANDOMIZE USR R
[illegible] RETURN
[illegible]
[illegible] REM Koniec listingu
```

Opis programu źródłowego w asemblerze

Wszystkie procedury wewnętrzne mają etykiety rozpoczynające się znakiem $. Wykorzystano ponadto dwie procedury zewnętrzne z ROM-u opisane w liniach 70,80. Pierwsza z nich testuje klawiaturę. Po wywołaniu w rejestrze E znajduje się wartość 255, jeśli nie wciśnięto żadnego klawisza lub wartość 0—39, jeśli klawisz został wciśnięty. W przypadku, gdy użyto ponadto klawisza funkcyjnego (CAPS lub SYMBOL SHIFT), jego kod znajdzie się w rejestrze D. W interesującym nas przypadku rejestry powinny zawierać D = 24 E = 32. Druga procedura rezerwuje miejsce w programie BASIC. Do pary HL wpisujemy adres pierwszego zarezerwowanego bajtu, do BC liczbę bajtów. Po jej wywołaniu program i zmienne zostają przesunięte do góry (w stronę wyższych adresów), a odpowiednie zmienne systemowe uaktualnione. W naszym przypadku rezerwowane jest miejsce za REM w pierwszej linii programu. Wykorzystywana jest tu zmienna systemowa PROG zawierająca adres pierwszej linii programu BASIC. Adres ten równy zwykle 23755 może ulec automatycznej zmianie w wypadku użycia niektórych urządzeń zewnętrznych, np. w przypadku korzystania z microdrive wynosi 23813.

Począwszy od adresu lokacji programu jest zarezerwowany obszar pamięci o długości 288 bajtów. Zwróćmy ponadto uwagę na to, że opisywane w liniach 990—1130 zmienne mają różną długość. Wszystkie zmienne opisujące parametry zleceń są jednobajtowe.

Program jest uruchamiany od adresu 65000. Po skoku do menu jest testowana wartość zmiennej

KODINS, po czym jest realizowany odpowiedni podprogram. Bliższego wyjaśnienia wymaga zlecenie zamiany obrazu realizowane przez podprogram $ZMIAN. Najpierw zostanie wyznaczony przez procedurę $USTPA adres zadanego wiersza w pamięci obrazów, a następnie wywoływana jest zasadnicza procedura $LINIA. Dokonuje ona zmian kolejnych wierszy ekranu.

Zamiana każdego wiersza odbywa się w trzech etapach. Najpierw wiersz ekranu jest przesyłany do bufora, następnie wiersz z pamięci jest przesyłany na ekran i na koniec wiersz z bufora do pamięci. W pierwszym i drugim etapie ze względu na nieliniową organizację ekranu wywoływane są procedury $ADRLI i $ADRAT wyznaczające adresy ekranowe początku wiersza i jego atrybutów. Bezpośredniej zamiany bajtów jednego wiersza dokonują procedury $ZAMLI i $ZAMAT. Kierunek przesyłania wyznacza zmienna WSKZAM. W trzecim etapie wiersz w buforze jest już uporządkowany i wystarczy przesłać go do pamięci rozkazem LDIR. Po dokonaniu zamian wszystkich wierszy adres aktualnego odczytu pamięci jest ustawiany z powrotem na początku zadanego wiersza. Kolejne wykonanie zlecenia zamiany z identycznymi parametrami spowoduje powrót do stanu wyjściowego na ekranie.

W przypadku użycia trybu przerwań 2 obsługę zamiany obrazów przejmie automatycznie procedura $PRZER. Oczywiście przed uruchomieniem tego trybu należy ustawić sensowne parametry zamiany. Naciśnięcie klawiszy SYMBOL SHIFT+SPACE spowoduje pierwszą zamianę obrazu, a ich puszczenie drugą. Wyjaśnienie tego zjawiska wymaga kilku dodatkowych informacji.

Tryby przerwań

Interpreter BASIC-a na Spectrum wykorzystuje tryb 1 przerwań. Jest on zawsze ustawiany po wykonaniu funkcji RESET. W trybie tym w momencie wystąpienia przerwania (impulsy takie generuje co 20 ms ULA) następuje skok do programu obsługi przerwania pod adresem 38H. Tutaj znajduje się podprogram obsługi, który między innymi testuje klawiaturę i odpowiednio modyfikuje zmienne systemowe (np. LAST-K). Dzieje się tak zawsze, o ile przerwania maskowalne są włączone (szczegóły w art. R. Wacławka „Zegar cyfrowy w ZX Spectrum").

Procedura obsługi przerwania przede wszystkim wyłącza przerwania, a następnie przechowuje na stosie rejestry i sprawdza, czy nastąpiło naciśnięcie interesujących nas klawiszy (SSHIFT+SPACE). Jeśli nie, to odtwarza rejestry i odsyła procesor do normalnej obsługi przerwania, a jeśli tak, to najpierw realizuje własne zadania. Po ich wykonaniu następuje skok do normalnej obsługi przerwania.

Od Redakcji:

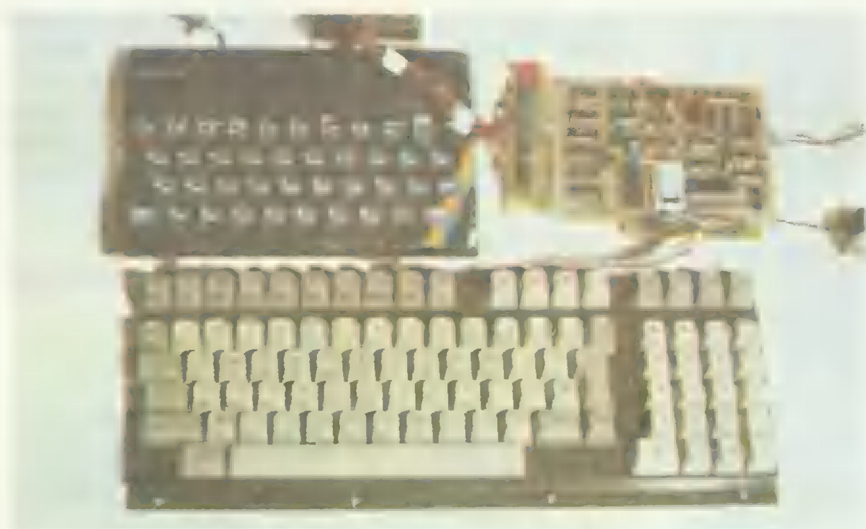
Drugi typ przerwań mikroprocesora Z-80 umożliwia zrealizowanie operacji bardzo trudnych lub niemożliwych do wykonania przy pomocy jedynie środków programowych. Przykładem może być opisany powyżej program zamiany obrazów, zupełnie nie ingerujący w oprogramowanie rezydujące aktualnie w pamięci komputera. Innym przykładem może być zrealizowanie bufora danych wprowadzanych z klawiatury — niektóre programy nie radzą sobie z dostatecznie szybkim przetwarzaniem informacji (np. popularny edytor Tasword) i przy zbyt szybkim jej wprowadzaniu następuje gubienie znaków. Niestety, w literaturze, nawet dość poważnej, znaleźć można wiele nieścisłości na temat mechanizmu pobierania adresu podprogramu obsługi przerwania w trybie 2. Przede wszystkim dotyczy to młodszego bajtu adresu, pod którym umieszczono młodszy bajt adresu programu obsługi przerwania (czyli wektora przerwania). W literaturze (np. praca zbiorowa „Modułowe systemy mikrokomputerowe") spotyka się stwierdzenia, że procesor pobiera w chwili potwierdzenia przerwania (w trybie drugim oczywiście) młodszy bajt adresu wektora przerwania z zaniedbaniem najmłodszego bitu — jest on jakoby ustawiany na zero. Tymczasem nie jest to prawda — procesor pobiera z linii danych pełny bajt, składany następnie z zawartością rejestru przerwania I. A zatem w przypadku ZX Spectrum nie wykorzystującego specjalnych (nietypowych) urządzeń zewnętrznych z linii danych, podpiętych do +5 V przez odpowiednie rezystory, odczytany zostanie bajt FFH, a nie FEH. Przykładowo, dla zawartości rejestru I równej 80H adres wektora przerwania umieścić należy w pamięci pod adresami 80FFH (młodszy bajt) i 8100H (starszy bajt).

W SPRAWIE ARTYKUŁU „PROBLEM COLLATZA A KOMPUTER"
Z 6 NUMERU „MŁODEGO TECHNIKA" Z ROKU 1986

Ponieważ otrzymałem od Państwa listy dotyczące problemu jaki przedstawiłem w szóstym numerze „Młodego Technika", a niestety, za co przepraszam, nie na wszystkie udzieliłem odpowiedzi, chciałbym uczynić to teraz. W większości Państwa listów, jeżeli nie wszystkich, znalazłem informacje o rozwiązaniu przez Państwa problemu Collatza, a jednocześnie pytają mnie Państwo co dalej.

Ponieważ to nie ja jestem twórcą tego problemu ani nie ja ufundowałem nagrodę za jego rozwiązanie chciałbym odesłać Państwa do artykułu, który szeroko o nim traktuje. Jest to: „American Mathematical Monthly" ze stycznia 1985 roku, „*The 3x+1 problem and its generalizations*" — J. C. Lagarias.

Adam Czeżowski



Jak z ZX SPECTRUM zrobić komputer?

część I

Grzegorz Zalot

Jak z ZX Spectrum zrobić komputer? Pytanie na pozór banalne, lecz każdy, kto trochę na komputerze tym popracował, zauważy w nim głębszy sens. Maszyna ta chcąc nie chcąc stała się najbardziej popularnym i najlepiej oprogramowanym mikrokomputerem domowym w Polsce, głównie z racji ceny, a także niemałej do niedawna popularności na Zachodzie. Wysłużone Spectrum ma wielu wrogów: zarzucają oni mu ograniczone możliwości sprzętowe, dużą awaryjność, niewygodną obsługę, a także znaczący wpływ na niewłaściwe ukierunkowanie komputerowej edukacji naszej młodzieży (jest ona jakoby bardziej zabawą, niż drogą do profesjonalnej informatyki). Wiele z tych argumentów rzeczywiście odpowiada stanowi faktycznemu, ale czy przez to koniecznie trzeba od razu sprzęt ten skierować na wysypiska śmieci? Nie wdając się tutaj w polemikę z zarzutami, co lub kto kieruje naszą młodzież na niewłaściwe tory mikrokomputerowej edukacji nie możemy przecież nie wziąć pod uwagę stanu faktycznego sytuacji sprzętowej w kraju. Stąd też na łamach „InforMika" kontynuujemy w tym momencie cykl materiałów opisujących różne przeróbki ZX Spectrum poprawiające jego możliwości funkcjonalne oraz komfort pracy — są one wynikiem doświadczeń ludzi, którzy chcą coś zrobić dla poprawy nie najlepszej przecież sytuacji w zakresie sprzętu. Wykonanie niektórych przeróbek wiąże się niestety z pewnym nakładem pracy, czasem trzeba uciec się do „prywatnego" importu podzespołów (tych najtańszych — korzystniejszych do zakupu za granicą) — są to niestety realia, których nie unikniemy (co jest możliwe w przypadku oprogramowania). Mimo wszystko uważam jednak, że warto włożyć trochę pracy w udoskonalenie naszych Spectrum — efekty powinny być całkiem niezłe. Przy okazji zachęcam wszystkich, którzy samodzielnie dokonali jakichś przeróbek w ZX Spectrum, do podzielenia się doświadczeniami z innymi — napiszcie, a ciekawy materiał z pewnością opublikujemy.

Część I
Budujemy profesjonalną klawiaturę

Klawiatura jest jednym z najważniejszych elementów mikrokomputera. Umożliwia ona dialog z maszyną, bezpośrednio wpływa na komfort pracy, szybkość zmęczenia, ilość popełnianych błędów itd. Niestety, praca z kompilatorami języków wyższego rzędu czy edytorami tekstu jest co najmniej utrudniona. Wielu posiadaczy Spectrum dobudowuje klawiaturę łączoną równolegle ze standardową wykonaną w oparciu o zestyki kontaktronowe — poprawia to nieco jej parametry pracy (głównie trwałość), lecz nie usuwa kłopotliwej sekwencji włączania trybu EXT, operowania klawiszami SHIFT itd. Poniżej opisujemy konstrukcję pozbawioną tych wad, wykorzystującą praktycznie każdą klawiaturę o zestykach ułożonych w matrycy do 16 x 6. Oryginalnie układ jest przystosowany do klawiatury będącej obecnie w RFN w wyprzedaży (cena ok. 10 DM), a pochodzącej z mało znanego modelu Commodore 610. Oczywiście sztywne przywiązanie do tej klawiatury nie jest konieczne — wykorzystanie innej (o innym ułożeniu klawiszy w matrycy) wymaga jedynie zmiany danych do zapisania w EPROM-ie. Klawiatura ta ponadto jest dołączana do każdego Spectrum bez dokonywania jego przeróbek, a nawet otwierania — układ łączymy z komputerem za pomocą złącza krawędziowego. Niestety, wiąże się to z niemałym stopniem skomplikowania układu, lecz chyba warto — osiągnięte efekty są naprawdę niezłe.

Na wstępie omawiania konstrukcji przypomnijmy, w jaki sposób w Spectrum jest dokonywany odczyt klawiatury. Klawiatura licząca 40 klawiszy jest połączona w matrycę 8 x 5. Oznacza to, że zestyk każdego z 40 klawiszy zawiera jedną z 8 linii wybierania tzw. półrzędu i jedną z 5 linii kolumn. Owych 8 linii wybierania półrzędów jest połączonych z ośmioma starszymi liniami adresowymi (A8 — A15; przez diody). Przyporządkowanie klawiszy jest takie, że każda linia półrzędu łączy 5 klawiszy leżących na lewo lub na prawo od środka każdego rzędu. Pozostałych 5 linii dekodujących stan klawiatury łączy po 8 klawiszy, po 1 z każdego półrzędu. Stan tych linii może być odczytywany przy pomocy portu o adresie 254 zawartym fizycznie w układzie ULA, na 5 młodszych bitach szyny danych. Przyporządkowanie linii jest takie, że klawiszom zewnętrznym każdej grupy odpowiada linia D0, idąc do wewnątrz D1 itd. do D4.

Przyporządkowanie półrzędów do linii adresowych jest następujące:

linia A8 — półrząd od CAPS SHIFT do V

— linia A9 — półrząd od A do G itd...

— linia A14 — półrząd ENTER do H

— linia A15 — półrząd od SPACE do B

Tak więc np. klawiszowi T odpowiada linia A10 i D4, klawiszowi K — A14 i D2. Każdorazowo dekodowany jest niski stan jednej z linii D0 — D4, gdyż w czasie odczytu klawiatury tylko jedna z linii A8 — A15 ustawiana jest na 0 i naciśnięcie któregokolwiek klawisza powoduje pojawienie się zera logicznego na określonej z 5 linii odczytu (układ ULA pełni w tym momencie tylko funkcję wzmacniaczy i bufora magistrali adresowanego jako urządzenie zewnętrzne o adresie 254).

Wiemy zatem już, jak jest dekodowany stan klawiatury — przy pomocy portu 254 oraz starszego bajtu adresowego. Można zatem bez większego trudu skonstruować układ, który na podstawie stanu linii A8 — A15 przy odczycie portu 254

poda na linię danych informację o naciśniętym klawiszu. Bliżej mechanizm ten poznamy już przy dokładniejszej analizie działania proponowanego układu.

Wspomnieliśmy na początku, że proponowany układ wykorzystuje w wersji modelowej klawiaturę zakupioną na wyprzedaży [illegible] RFN. Klawiatur tych jest obecnie w Polsce już sporo, [illegible] dla informacji: kupić je można w RFN w sklepach firmy Volkner. Klawiatura ta posiada 94 klawisze działające na zasadzie zwierania ścieżek za pomocą kształtki z gumy przewodzącej. Klawisze te (identyczne, jak w Commodore C16 i C64) mają dobre prowadzenie i są bardzo trwałe. Zestyki klawiszy ułożone są w matrycę 16 × 6 — rys. 1. Dużą zaletą klawiatury tej jest wydzielony blok numeryczny, klawisze sterowania kursorem, trybem pracy komputera oraz 10 klawiszy funkcyjnych — wykorzystując dwa klawisze SHIFT (normalny i CTRL — u nas: funkcyjny) można bez większego problemu na klawiszach umieścić wszystkie funkcje i znaki ZX Spectrum wywoływane we wszystkich trybach. Dzięki specjalnej konstrukcji interfejsu klawiatury nie jest konieczne stosowanie sekwencji EXT + klawisz — funkcja EXT jest generowana automatycznie.

Schemat ideowy układu klawiatury i interfejsu jest przedstawiony na rys. 2 a – c. Analizę rozpoczniemy od fragmentu z rys. 2 b.

Podstawowym blokiem jest zespół licznika i układu tzw. przeglądania matrycy klawiatury. Licznik zlicza modulo 96 (czyli 16 × 6), przy czym młodsze 4 bity wybierają jedną [illegible] 16 kolumn, a starsze 3 — jeden z 6 wierszy. W skład licznika wchodzi układ 7493 (US21) oraz trzy 7472 (US25–27). Dekoder 74154 (US25—27) wybiera poszczególne kolumny poprzez diody oddzielające, natomiast odczyt wiersza jest dokonywany poprzez wzmacniacze (wtórniki) W1– W6 i multiplekser 74151 (US19). W stanie spoczynkowym (nie jest naciśnięty żaden klawisz) wiersze mają potencjał wysoki (rezystory 47 kOm do + 12 V). Po naciśnięciu klawisza przy wybraniu odpowiedniej kolumny jeden z wierszy przyjmuje stan niski, odczytywany następnie przy odpowiednim stanie starszych 3 bitów licznika (pełny cykl licznika powtarzany jest ok. 60 razy na sekundę). Przy odczycie linii tej zero logiczne jest podane przez układ opóźniający na wejście bramki B17.3 (pierwsza liczba określa nr układu scalonego, druga nr bramki lub przerzutnika w układzie) i dalej po negacji odblokowuje monowibrator M13.2. Monowibrator ten jest wyzwalany następnie przez sygnał [illegible] generatora (B27.1, B27.2, B20.4) tak długo, dopóki nie ustąpi jedynka [illegible] jego wejścia B (jest to monowibrator z tzw. podtrzymaniem). Jednocześnie zablokowany zostaje licznik (bramka B17.2). Układ różniczkujący na wejściu bramki B20.3 likwiduje możliwość zmiany stanu licznika w chwili jego zablokowania.

Impulsy wyzwalające monowibrator są podawane przez bramkę B17.4 — istnieje możliwość ich zablokowania przez układ bramek podłączony do jej drugiego wejścia. Blokada ta jest wymagana dla zdekodowania stanu klawiszy SHIFT i FUN (oryginalny CTRL), naciskanych jednocześnie [illegible] innymi klawiszami — w tym przypadku nie można przecież blokować dalszego przeglądania klawiatury. W proponowanym układzie klawiszy nie przerywających przeglądania klawiatury jest 5 — klawisz SHIFT, FUN, CAPS SHIFT, SYMBOL SHIFT (dwa ostatnie sterują bezpośrednio funkcjami klawiatury Spectrum) oraz Q (w niektórych programach jest wykorzystywany jako dodatkowy SHIFT, tu z możliwością wyłączenia blokady). Blokada występuje wtedy, gdy zostanie zdekodowana koincydencja wyboru określonego wiersza i kolumny klawiatury. Poszczególne sygnały blokady są sumowane „na drucie" za pomocą bramek typu *Open Collector* (w tym jedna przerobiona ze standardowej za pomocą diody).

Klawisze nie blokujące przeglądania klawiatury muszą jednak być jakoś dekodowane — do tego celu służy fragment na rys. 2c. Poszczególne przerzutniki (typu D — 74) są strobowane sygnałami

Rys.1 ROZKŁAD POŁĄCZEŃ KLAWIATURY COMMODORE 610

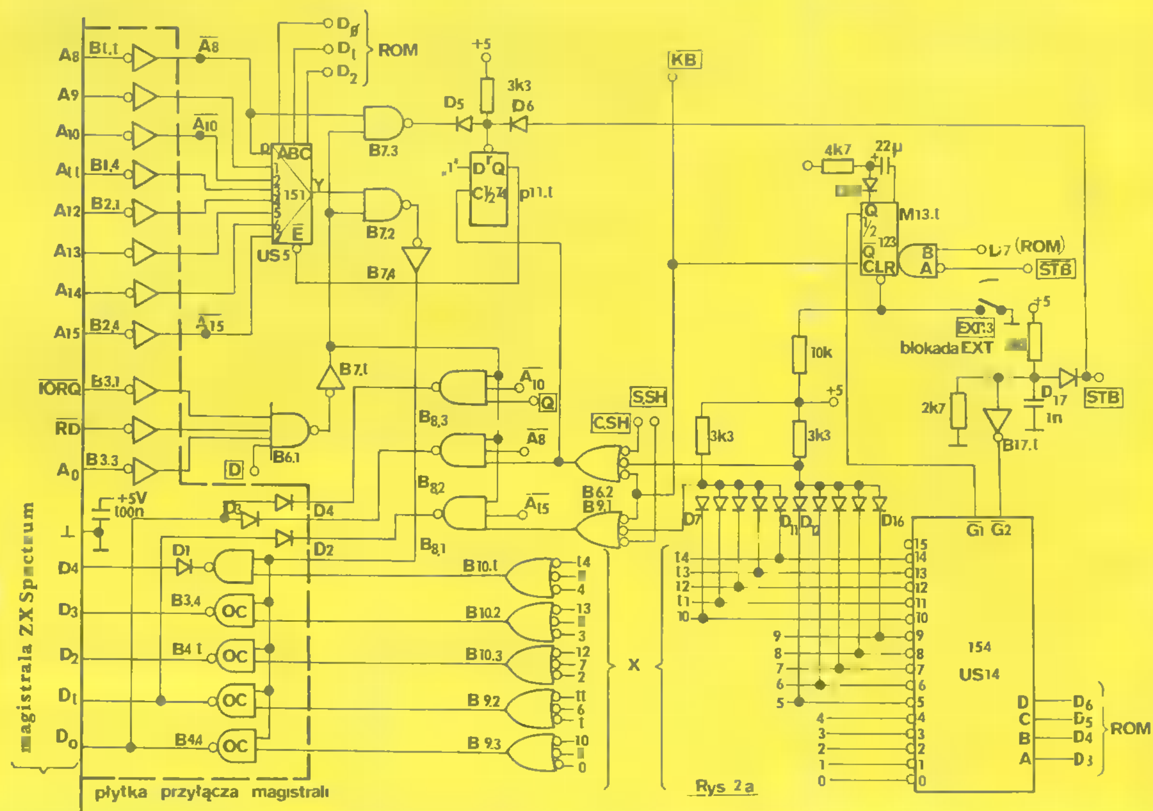
	1	[illegible]	3	[illegible]	[illegible]	[illegible]	7	[illegible]	9	10	11	12	13	14	15	16
17	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	[illegible]	[illegible] *	CLR HOME	RVS OFF	NORM GRAPH	RUN STOP
18	ESC	1	2	3	4	5	7	8	9	0	±	← *	? *	CE *	* *	/ *
19	TAB	Q	W	E	R	6	U	I	O	—	←	→	7	8	9	— *
[illegible]	n.c.	A	S	D	T	Y *	J	K	L	P	]	INS DEL	4	5	6	+ *
21	SHIFT	Z *	X	F	G	H	M	,	;	[	RET	C= *	1	2	3	ENTER
[illegible]	CTRL	n.c.	C	V	B	N	SPACE	•	/	"	π	n.c.	0	•	00 *	n.c.

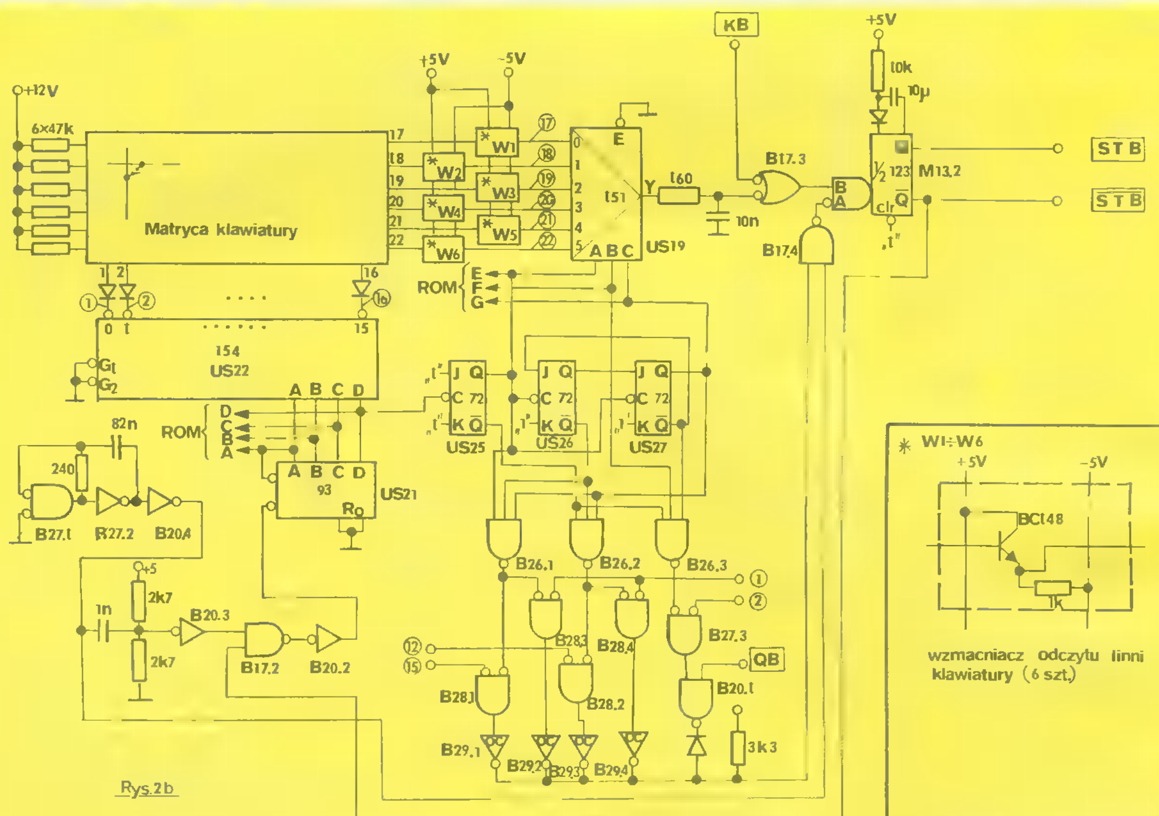
łączówka:

[illegible] 10 11 12 13 14 15 16 1 [illegible] 22 21 20 19 18 17

* – zmiana znaczenia klawisza [illegible] wykonania modelowego

n.c. – niewykorzystany styk


magistrala ZX Spectrum
płytka przyłącza magistrali
ROM
KB
blokada EXT
Rys 2a

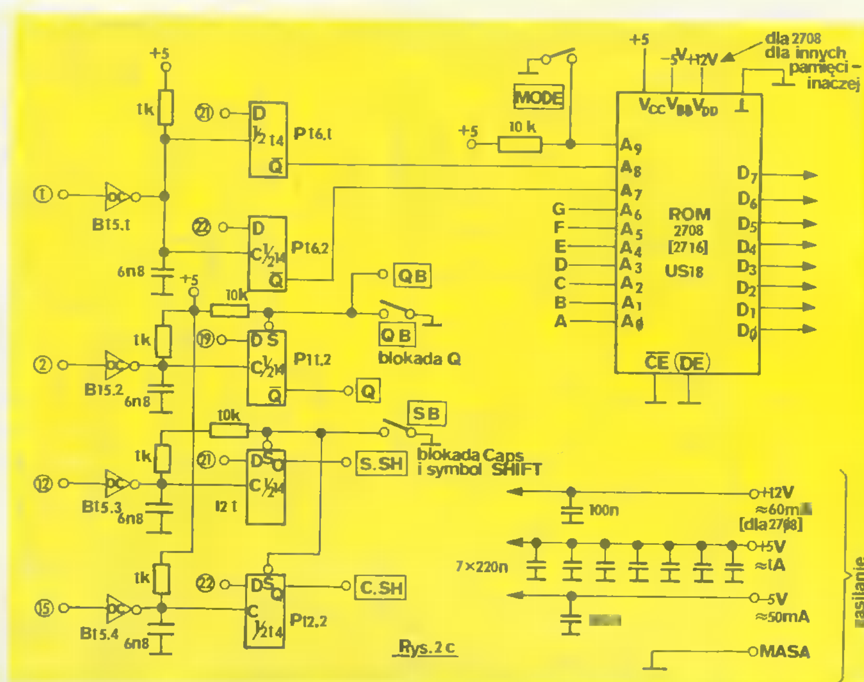

Matryca klawiatury
ROM
KB
STB
WI÷W6
wzmacniacz odczytu linni klawiatury (6 szt.)
Rys.2b

wyboru kolumny, z niewielkim opóźnieniem koniecznym do kompensacji czasu propagacji przez linie klawiatury i wzmacniacze. Dane dla przerzutników są pobierane z wyjść wzmacniaczy wiersza klawiatury. Na schemacie widać przełączniki blokujące działanie klawiszy CAPS i SYMBOL SHIFT oraz Q (normalnie powinien być zablokowany!). Natomiast wyjścia sygnałów SHIFT i FUN SHIFT są podane bezpośrednio na wejście adresowe pamięci ROM, razem ze stanem licznika (odpowiada kodowi naciśniętego klawisza) i stanem dodatkowej linii MODE, pozwalającej wykorzystać ewentualnie inne przyporządkowanie funkcji do klawiszy.

Siedem bitów podanych na pamięć ROM określa numer naciśniętego klawisza poszerzony o dwa bity informacji o SHIFT-ach. A zatem jeżeli przyjąć, że może być naciśnięty jeden z SHIFT-ów lub żaden, to otrzymujemy trzy możliwe do zakodowania znaki przypadające na jeden klawisz (naciśnięcie dwóch SHIFT-ów oraz trzeciego klawisza może w niektórych sytuacjach wywołać niejednoznaczność odczytu stanu klawiatury — nie dotyczy to CAPS SHIFT i SYMBOL SHIFT. Na każdy znak przypada w tym przypadku jeden bajt dostępny na wyjściu ROM-u — jest on dekodowany w układzie przedstawionym na rys. 2a.

Do właściwego podania informacji na linie danych potrzebnych jest kilka informacji: sygnał odczytu portu 254, numer wyzerowanej linii starszego bajtu adresowego, numer linii danych, którą należy wyzerować oraz informacja o naciśniętym SHIFT-cie (oczywiście SHIFT-cie Spectrum, czyli CAPS lub SYMBOL). Sygnał odczytu portu 254 generuje bramka B6.1 (bramki B1.x, B2.x i B3.x spełniają funkcję buforów — są one typu LS — a ponadto umożliwiają odsunięcie interfejsu od magistrali na odległość większą od kilkunastu cm). Numer wyzerowanej linii starszego bajtu adresowego, przy której należy wystawić dane na linie danych, wybiera multiplekser US5, na podstawie stanu 3 najmłodszych bitów słowa pamięci ROM — bity te kodują zatem numer półrzędu klawiatury Spectrum. W przypadku zgodności otwierane są bramki OC B4.1—B4.4 i B3.4 (z diodą).

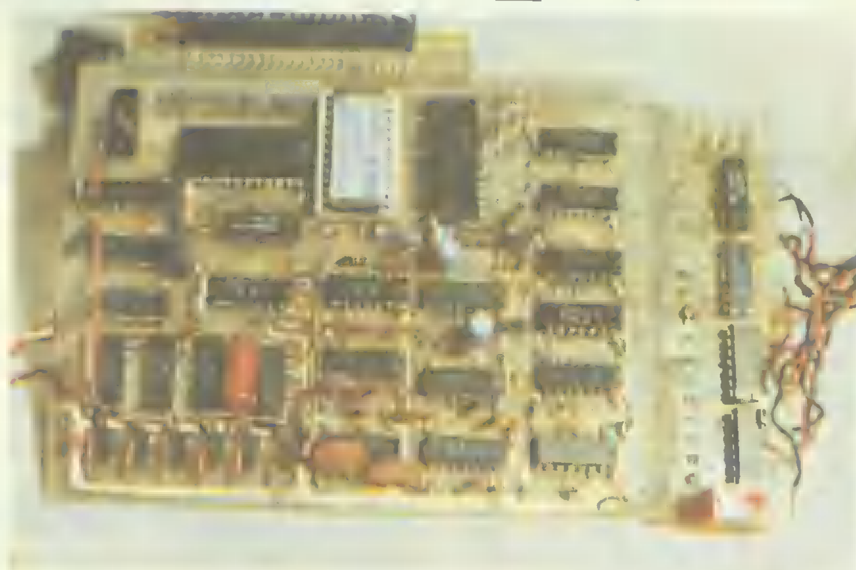
Kolejne 4 bity słowa pamięci ROM kodują numer klawisza w półrzędzie — jest ich w każdym 5. Dodatkowo należy jednak podać informację o naciśnięciu SHIFT-ów. Zrealizowano to w ten sposób, że liczba kodowana przez owe 4 bity z przedziału 0—4 koduje klawisz bez SHIFT-ów, z przedziału 5 — 9: z CAPS SHIFT, a z przedziału 10—14: z SYMBOL SHIFT (stan 15 określa brak naciśnięcia klawisza). W ten sposób optymalnie wykorzystano te 4 bity. Realizacja dekodowania jest przeprowadzona w oparciu o dekoder US14 i zestaw bramek (B10.x i B9.2 3) oraz zespół diod generujących sygnał odpowiadający odpowiedniemu SHIFT-owi (diody D7—

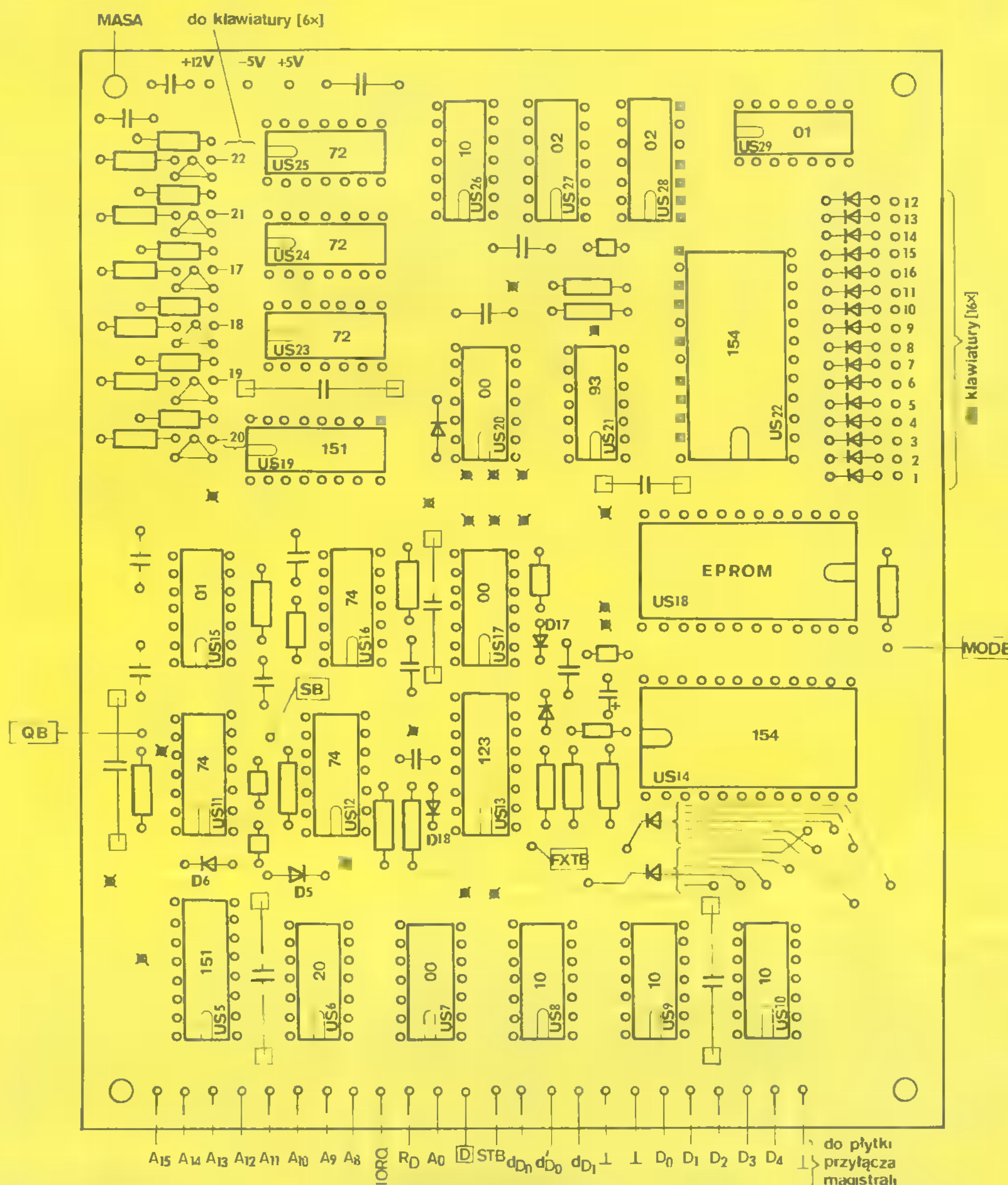


—D16). W stanie spoczynkowym dekoder nie jest odblokowany (sygnały G1 i G2) — uruchamiany jest dopiero w momencie odczytu klawisza (sygnał z monowibratora M13.2 — oznaczony STB). Dopiero w tym momencie jest uaktywnione określone wyjście dekodera i podana informacja na linie danych. W przypadku niewykorzystania funkcji koduje się stan 15 — nie wywołuje to żadnej zmiany na magistrali danych. Układ opóźniający z diodą D17 zapewnia kompensację czasu dostępu pamięci ROM i pozostałych układów.

Pozostał nam jeszcze jeden bit słowa pamięci ROM — koduje on konieczność wygenerowania sygnału EXT (bit D7). W przypadku jego ustawienia uruchamiany jest monowibrator M13.1, blokujący klawiaturę na dodatkowy okres (ok. 30 ms) i wystawiający na szynę danych oba SHIFT-y jednocześnie. Na ten czas musi być oczywiście zablokowany dekoder US14 (wejście G1). Ze względu na fakt, że wiele kompilatorów języków wyższego rzędu nie akceptuje sygnału EXT, wprowadzono przełącznik blokujący jego automatyczne wystawianie — oznaczony EXT B.

Oba klawisze SHIFT w ZX Spectrum oraz litera Q są generowane przy pomocy bramek B8.x — wykrywają one fakt równoczesnego wybrania odpowiedniej linii starszego bajtu adresowego, zaadresowania interfejsu oraz strobu danego sygnału i przez diody D2—D4 wymuszają zero na odpowiedniej linii. Bramki B6.2 i B9.1 sumują żądania wystawienia SHIFT-ow



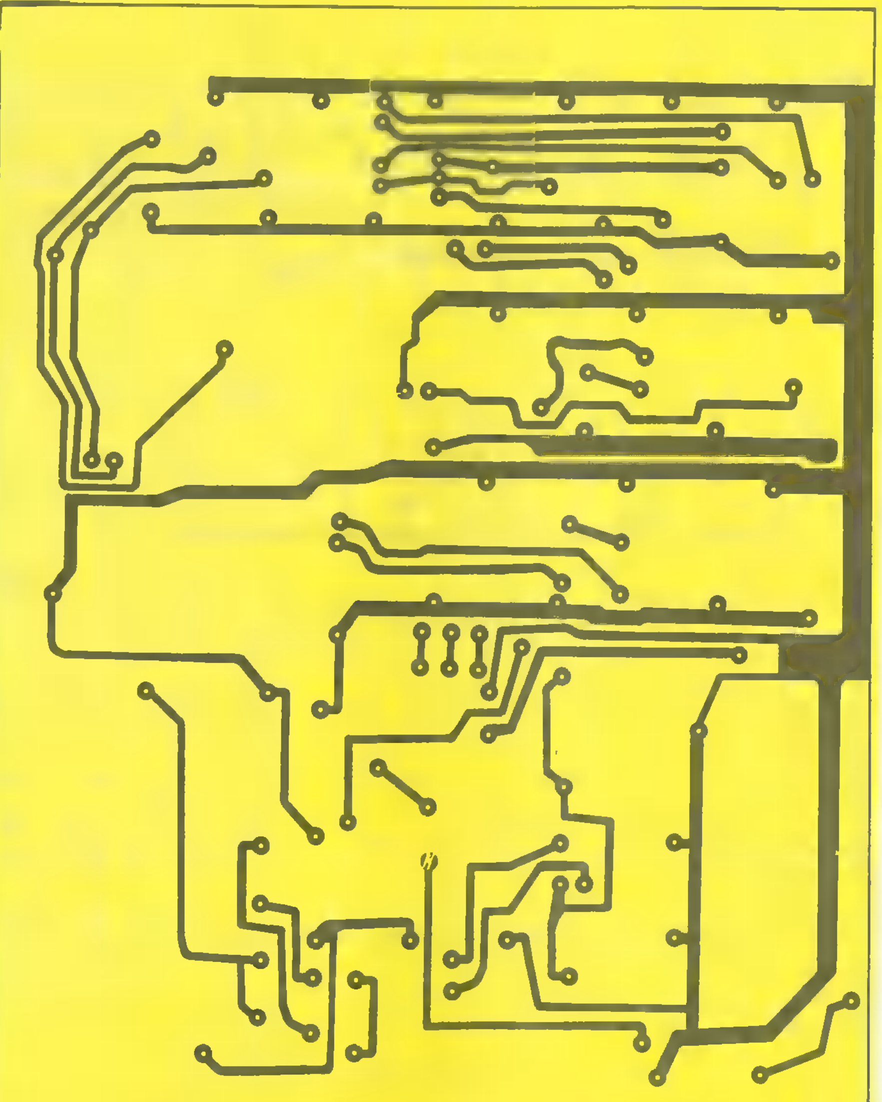


Rys.3a PŁYTKA DRUKOWANA – ROZMIESZCZENIE ELEMENTÓW – WIDOK OD STRONY MONTAŻU
WYMIARY PŁYTKI: 147,5 × 117 mm
■ — łączenie ścieżek strony montażu i strony lutowania
—kondensatory blokujące napięcie 5 V lutowane bezpośrednio do ścieżek na płytce

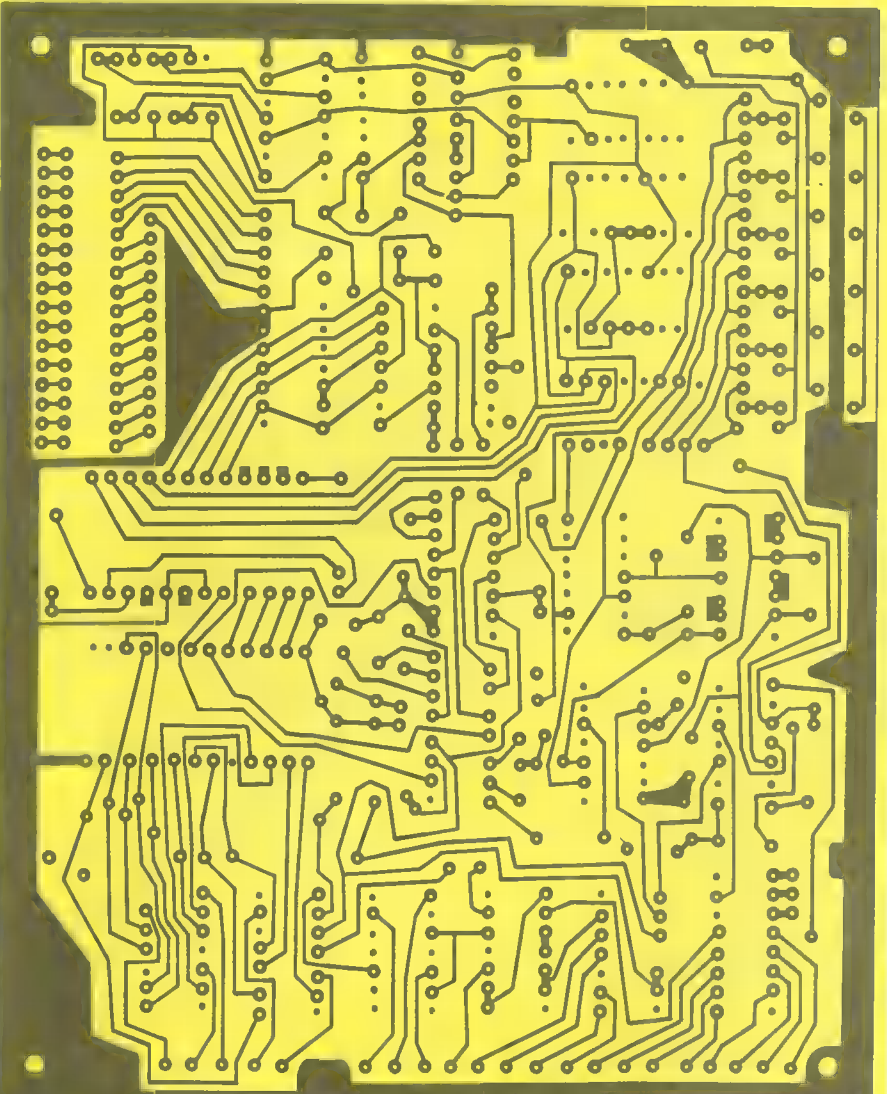
od układu generowania EXT, klawiszy Caps SH i Symbol SH oraz dekodera wspomnianych 4 bitów (klawisz z SHIFT lub bez).

Kilka słów ■ temat elementu P11.1, bramki B7.3 i towarzyszących diod. Otóż procedura odczytu klawiatury w ROM-ie Spectrum sprawdza półrzędy od 0 do 7 (adresy od A8 do A15). Oznacza to, że w pierwszym cyklu rozpoznany zostaje m.in. CAPS SHIFT, ■ w ostatnim SYMBOL SHIFT. Jeżeli zatem interfejs wystawi informację po odczytaniu przez mikroprocesor CAPS SHIFT-u (czyli po pierwszym cyklu odczytu), ■ przed odczytem odpowiedniego (innego) półrzędu, to dla komputera będzie to odczyt klawisza bez SHIFT-u. Sytuacja taka może wystąpić jedynie dla CAPS SHIFT, gdyż SYMBOL SHIFT jest wykrywany zawsze na końcu procedury. Aby zapobiec takiemu zjawisku (występuje ono dla ok. co 100—200 naciśnięcia klawisza ■ CAPS SHIFT), wprowadzono przerzutnik blokujący dekoder US5 (P11.1). Jest on ustawiany (co blokuje dekoder) przy wystawieniu SHIFT, a zerowany po odczycie SHIFT (linia adresowa A8) lub w stanie spoczynkowym klawiatury (zerowy sygnał STB). Takie jego działanie powoduje, że zawsze najpierw rozpoznany zostanie CAPS SHIFT, a dopiero potem właściwy klawisz. Rozwiązanie to ■ jedną drobną wadę: w niektórych sytuacjach nie działa klawisz BREAK — wtedy, gdy nie jest testowany SHIFT, czyli w operacji LOAD lub SAVE. Należy w tych przypadkach zamiast BREAK używać klawisza SPACE.

Czas teraz na opis programowania pamięci ROM — w interfejsie najlepiej zastosować jedną ■ mniejszych pamięci EPROM, np. wycofane już niemal zupełnie 2708 lub 2716. Dzięki dość znacznemu rozbudowaniu układu wykorzystujemy bardzo efektywnie pamięć ROM — inne rozwiązania, nie zawierające tych elementów, wymagałyby już pamięci o dłuższym słowie (czyli dwa elementy) względnie zaprogramowania znacznie większej ilości komórek — a to przy programowaniu ręcznym znacznie utrudnia wykonanie interfejsu, ponadto zwiększa prawdopodo-

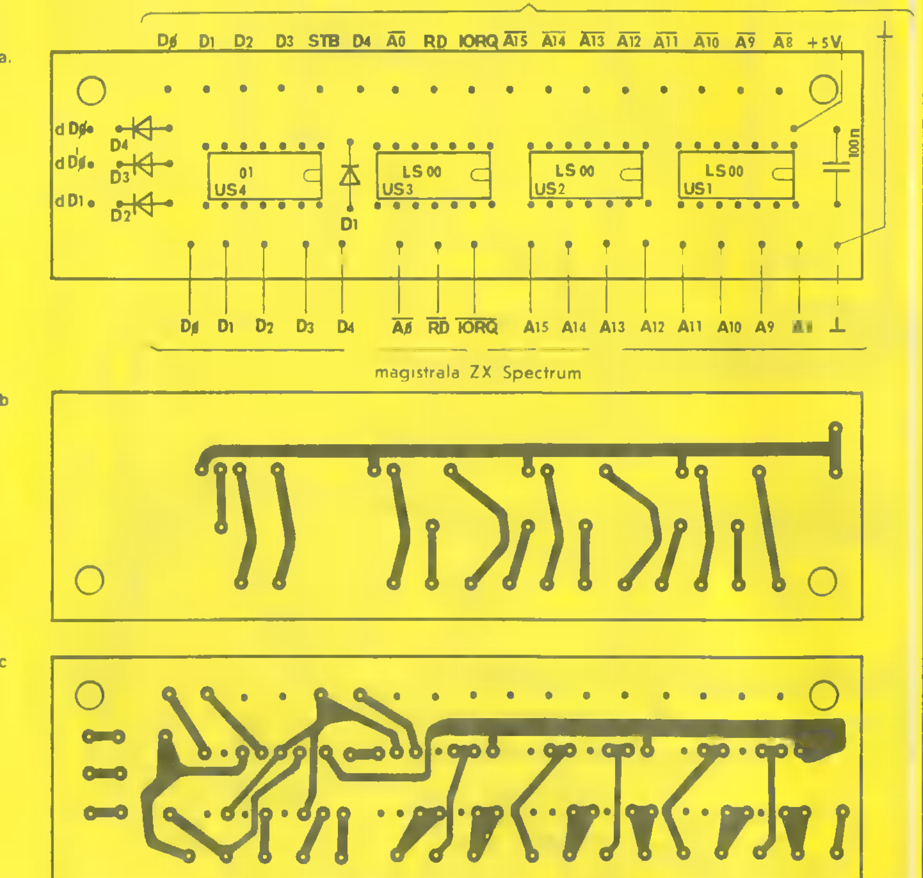


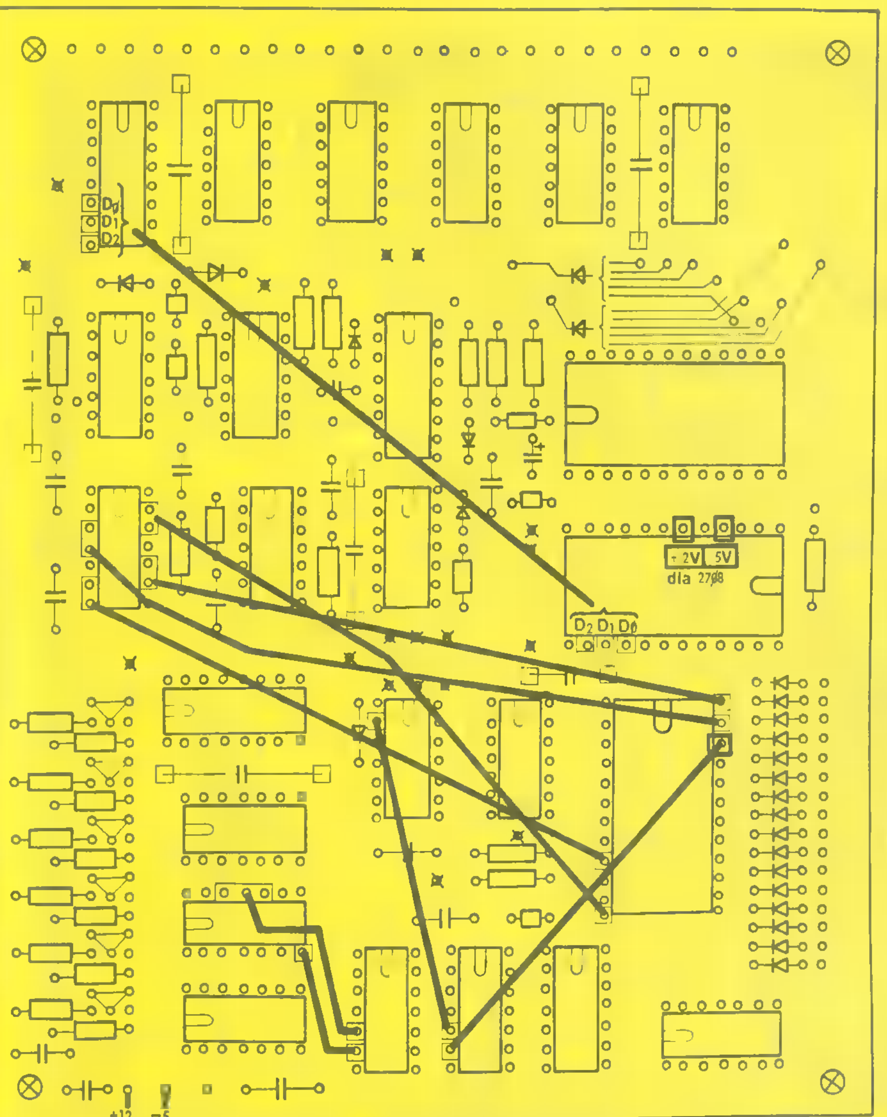
Rys.3c PŁYTKA DRUKOWANA–STRONA ELEMENTÓW (WIDOK OD STRONY ELEMENTÓW)



Rys.3b PŁYTKA DRUKOWANA–STRONA LUTOWANIA (WIDOK OD STRONY LUTOWANIA)

Rys.4. PŁYTKA PRZYŁĄCZA [redacted] – WYMIARY 107×30,5 mm
a) rys. montażowy – widok [redacted] strony elementów
b) strona elementów
c) strona lutowania





Rys.3d PŁYTKA DRUKOWANA – WIDOK [redacted] LUTOWANIA – POŁĄCZENIA PRZEWODEM [CYNA]

bieństwo popełnienia pomyłki. Adres bajtu kodującego zadaną funkcję klawisza określa położenie tego klawisza w matrycy klawiatury (wynika z połączeń) oraz naciśnięcie odpowiedniego SHIFT-u — te dane można wyznaczyć z łatwością. Co dalej? Określamy położenie klawisza, którego funkcję chcemy zrealizować, na klawiaturze Spectrum: jest to numer półrzędu, numer kolumny, rodzaj SHIFT-u oraz ewentualnie konieczność włączenia EXT. Poszczególne bity słowa pamięci ROM kodujemy natomiast następująco:

D7	D6 D5 D4 D3	D2 D1 D0
EXT	kod klawisza	numer półrzędu

gdzie: numer półrzędu — od 0 do 7 dla wyboru linii odpowiednio przez A8 do A15

kod klawisza — nr w półrzędzie licząc od 0 do 4, od klawisza najbardziej zewnętrznego do środka, dla klawisza z CAPS SHIFT dodać 5, dla SYMBOL SHIFT — 10

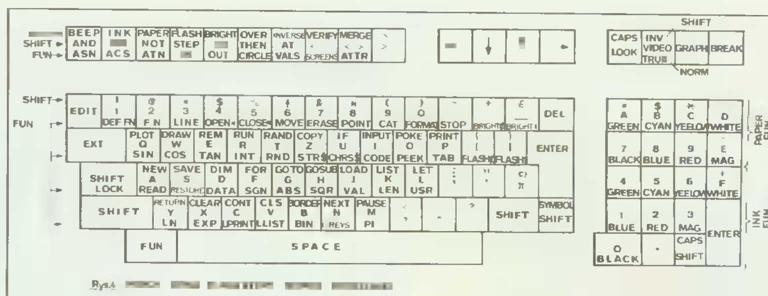
EXT — bit równy 1 dla konieczności włączenia trybu EXT, w przeciwnym wypadku 0

Procedurę tę najlepiej prześledzić na przykładzie: Np. chcemy zakodować klawisz K — półrząd 6, nr w półrzędzie — 2, bez EXT, czyli słowo będzie miało postać: 00010110 = 16H

Dla funkcji STEP — półrząd 1, nr klawisza — 2, z SYMBOL SHIFT, bez EXT: 01100001 = 61H

Dla OPEN# — półrząd 3, nr 3 z SYMBOL SHIFT i z EXT: 11101011 = EBH

Niestety, procedura ta jest trochę pracochłonna, ale można zaprogramować dowolne funkcje klawiszy, stosownie do potrzeb. W modelu przyjęto układ klawiszy, jak na rys. 5 — dla bloku ASCII zachowano normalne działanie klawisza SHIFT, dodatkową funkcję wywołuje się naciskając FUN. Klawisze funkcyjne mają po trzy funkcje: pierwsza od góry bez SHIFT, druga z SHIFT, trzecia z FUN. Sterowanie kursorem, tryb EXT, EDIT, GRAPH itd. działa niezależnie od SHIFT-ów. W bloku numerycznym wprowadzono duże litery od A do F (cyfry heksadecymalne), na SHIFT-cie znaki #, $, [illegible] z FUN następuje ustawianie kolorów



Rys.4

PAPER i INK. Dwa klawisze wydzielono do bezpośredniego sterowania CAPS i SYMBOL SHIFT Spectrum. Dla tego układu klawiszy i klawiatury z Commodore 610 zamieszczona obok tablica zawiera dane zapisane w pamięci ROM — są one podane heksadecymalnie.

Znamy już działanie układu, można przystąpić do omówienia konstrukcji. Na początku jednak jedna bardzo ważna uwaga. Otóż opisany układ interfejsu klawiatury nie należy do łatwych w budowie i uruchomieniu. **Porywanie się na jego budowę bez odpowiedniego przygotowania fachowego i zaplecza jest bezcelowe, szkoda czasu i pieniędzy na elementy.** Niełatwe jest także wykonanie dwustronnej płytki drukowanej, a uruchomienia może się podjąć jedynie zaawansowany elektronik znający dobrze układy cyfrowe.

Układ elektroniczny jest zmontowany na dwóch płytkach drukowanych. Główna zawiera wszystkie elementy poza buforami magistrali komputera — przedstawiona jest na rys. 3. Na rys. 3a pokazano rozmieszczenie elementów, na rys. 3b — druk od strony lutowania, rys. 3c — od strony elementów i na rys. 3d — połączenia, których wykonanie na druku wymagałoby znacznie większej precyzji i zrealizowano je za pomocą CYNAR-u. Druga płytka, zawierająca elementy współpracujące z magistralą mikrokomputera, jest przedstawiona na rys. 4. Konieczność zastosowania dwóch płytek wynika z faktu, że do magistrali komputera, posiadającej bardzo małą obciążalność, nie można dołączyć przewodów dłuższych, niż ok. 10 cm — mogłoby to spowodować zakłócenia pracy komputera. Połączenia płytki małej z główną mogą natomiast mieć już długość nawet 30—40 cm, pod warunkiem zastosowania tzw. przewodu taśmowego. Podłączenie do magistrali ZX Spectrum wykonujemy za pomocą odpowiedniego złącza krawędziowego, zgodnie z opisem styków zawartym w instrukcji komputera. Podłączenie to powinno być pewne, wykluczyć należy możliwość zwarć czy przerwy. Konieczne może być także wykonanie złącza przejściowego przy korzystaniu z innych interfejsów — jego konstrukcję pozostawiamy już jednak Czytelnikom.

[illegible]	[illegible]	Adres	Dane	Adres	[illegible]
0000	75	0020	37	[illegible]	7F
0001	6D	0021	[illegible]	0041	25
0002	59	0022	0A	[illegible]	10
0003	61	0023	12	0043	[illegible]
[illegible]	69	0024	1A	0044	[illegible]
[illegible]	71	0025	24	0045	[illegible]
[illegible]	[illegible]	0026	1D	[illegible]	17
[illegible]	[illegible]	0027	15	0047	6F
[illegible]	5A	0028	0D	0048	5D
[illegible]	62	0029	6E	[illegible]	[illegible]
000A	48	002A	54	004A	06
[illegible]	[illegible]	002B	[illegible]	004B	7F
[illegible]	[illegible]	[illegible]	1C	004C	03
[illegible]	[illegible]	002D	14	[illegible]	0B
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	002F	3A	[illegible]	06
[illegible]	[illegible]	0030	7F	0050	[illegible]
[illegible]	[illegible]	[illegible]	01	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	20
0014	1B	[illegible]	[illegible]	[illegible]	[illegible]
0015	[illegible]	[illegible]	[illegible]	0055	1F
0016	[illegible]	[illegible]	1E	[illegible]	[illegible]
0017	14	[illegible]	16	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	0058	[illegible]
0019	04	0039	05	[illegible]	[illegible]
001A	5E	003A	ED	005A	[illegible]
001B	44	003B	2C	005B	7F
001C	[illegible]	003C	1B	005C	04
001D	[illegible]	003D	23	005D	67
001E	40	[illegible]	24	[illegible]	7F
001F	39	003F	[illegible]	005F	7F
[illegible]	[illegible]	00A0	[illegible]	00C0	7F
[illegible]	E0	00A1	2A	00C1	4D
[illegible]	[illegible]	[illegible]	32	[illegible]	38
[illegible]	F0	00A3	3A	[illegible]	41
0084	F7	00A4	42	00C4	49
[illegible]	EF	00A5	76	00C5	4E
[illegible]	[illegible]	00A6	45	[illegible]	3F
[illegible]	EA	00A7	3D	00C7	6A
[illegible]	F2	00A8	35	00C8	58
[illegible]	E1	00A9	D1	[illegible]	E9
008A	4B	00AA	[illegible]	00CA	06
[illegible]	[illegible]	00AB	3C	00CB	7F
[illegible]	[illegible]	00AC	1C	00CC	03
[illegible]	[illegible]	00AD	14	00CD	0B
[illegible]	[illegible]	00AE	[illegible]	[illegible]	13
[illegible]	[illegible]	00AF	[illegible]	00CF	06
[illegible]	2B	00B0	7F	00D0	7F
[illegible]	[illegible]	[illegible]	[illegible]	00D1	7F
[illegible]	[illegible]	00B2	31	00D2	40
[illegible]	[illegible]	[illegible]	39	[illegible]	48
[illegible]	[illegible]	00B4	4A	[illegible]	4F
[illegible]	[illegible]	00B5	30	00D5	47
[illegible]	[illegible]	[illegible]	46	[illegible]	07
[illegible]	[illegible]	00B7	[illegible]	00D7	72
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	55
009A	[illegible]	[illegible]	F1	00DA	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	00DB	7F
[illegible]	[illegible]	00BC	1B	[illegible]	04
[illegible]	[illegible]	[illegible]	[illegible]	00DD	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	7F
009F	[illegible]	00BF	66	00DF	7F
0100	[illegible]	[illegible]	[illegible]	0140	7F
[illegible]	[illegible]	0121	[illegible]	0141	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	0142	90
[illegible]	[illegible]	[illegible]	[illegible]	0143	99
[illegible]	DD	0124	9A	0144	A1
[illegible]	F6	[illegible]	F4	0145	A6
[illegible]	EE	0126	9D	0146	[illegible]
[illegible]	E6	[illegible]	95	0147	7F
[illegible]	DE	0128	8D	0148	7F
[illegible]	D9	0129	51	[illegible]	BC
010A	4B	[illegible]	8C	[illegible]	06
010B	[illegible]	012B	[illegible]	[illegible]	7F
[illegible]	[illegible]	[illegible]	84	[illegible]	AB
[illegible]	[illegible]	[illegible]	83	014D	[illegible]
[illegible]	[illegible]	012E	8B	014E	BB
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
0110	[illegible]	0130	[illegible]	[illegible]	7F
0111	[illegible]	0131	81	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	89	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	91	[illegible]	A0
[illegible]	[illegible]	0134	A2	[illegible]	A7
[illegible]	F3	[illegible]	A5	[illegible]	9F
0116	[illegible]	0136	[illegible]	[illegible]	[illegible]
0117	E4	0137	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	8E	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	85	[illegible]	7F
011A	94	013A	B4	015A	7F
011B	[illegible]	[illegible]	[illegible]	[illegible]	7F
011C	[illegible]	[illegible]	C3	[illegible]	[illegible]
011D	A3	013D	[illegible]	[illegible]	[illegible]
[illegible]	A4	013E	CC	015E	[illegible]
[illegible]	9C	013F	[illegible]	[illegible]	[illegible]

WYKAZ ELEMENTÓW

1 Układy scalone LS00 — 3 szt.
00 — 3 szt.
01 — 3 szt.
02 — 2 szt.
10 — 4 szt.
20 — 1 szt.
72 — 3 szt
74 — 3 szt
93 — 1 szt
123 — 1 szt.
151 — 2 szt
154 — 2 szt
EPROM 2708, 2716 lub inny — 1 szt.
2. Tranzystory — BC148 lub podobne — 6 szt.
3 Diody — BAP795 lub podobne — 36 szt.
4. Rezystory
160 — 1 szt.
240 — 1 szt.
1k — 10 szt
2k7 — 3 szt.
3k3 — 5 szt.
4k7 — 1 szt.
10k — 5 szt.
47k — 6 szt.
5 Kondensatory: blokujące zasilanie
100—470 nF/100 V MKSE — 10 szt.
1n — 2 szt.
6n8 — 4 szt.
10n — 1 szt.
82n — 1 szt.
10 µF/25 V — 1 szt.
22 µF/25 V — 1 szt
6 Mikroprzełączniki — stabilne — 4 szt.
7 Klawiatura — wg opisu
8 Inne materiały — wg opisu i indywidualnych potrzeb

UWAGA!
Prawa autorskie niniejszej konstrukcji są zastrzeżone — produkcja zarobkowa wymaga pisemnej zgody autora!

Główna płytka posiada dodatkowe wejście oznaczone „D", podłączone do dekodera adresu. Służy ono do ewentualnego dołączenia pełnego układu dekodowania, w miejsce niepełnego, stosowanego typowo w Spectrum.

Na schematach celowo pominięto opisy niektórych elementów, numery wyprowadzeń układów scalonych itp. Ideą było wręcz zmuszenie potencjalnych wykonawców interfejsu do dokładnego przeanalizowania schematu, tak ideowego, jak i montażowego — tylko w ten sposób unika się błędów wynikających z przekłamań w druku, niewyraźnych rysunków itp. A zatem najpierw dokładnie sprawdzamy wszystkie schematy i rysunki z pomocą katalogu, a dopiero potem przystępujemy do gromadzenia elementów i budowy. Trzeba też będzie niestety znaleźć dojście do programatora EPROM-ów, względnie wykonać go samodzielnie, choćby w najprostszej wersji (por. art. K. Wiśniewskiego).

Całość zasilana musi być z oddzielnego zasilacza — układy TTL pobierają dość znaczny prąd (rzędu 1 A). Wymagane są ponadto dodatkowe napięcia −5 i +12 V. Korzystne byłoby zastosowanie wspólnego dodatkowego zasilacza dla interfejsu klawiatury i komputera — eliminując bardzo zawodny zasilacz fabryczny i przetwornicę można dodatkowo poprawić niezawodność pracy naszego Spectrum. Należy jedynie pamiętać o zabezpieczeniu zasilacza przed zanikiem −5 V (dopóki go nie ma, nie wolno włączyć +12 V). Zasilanie pamięci EPROM należy podłączyć odpowiednio do jej typu, podobnie linie adresowe. Starannie należy też wykonać połączenia przewodami (rys. 3d) oraz blokowanie zasilania kondensatorami bezindukcyjnymi — najlepiej typu MKSE.

Uruchamianie rozpoczynamy od układu przeglądania klawiatury. Sprawdzamy wzbudzanie się generatora, pracę liczników, zatrzymywanie zliczania po naciśnięciu klawisza z wyjątkiem klawiszy wydzielonych (SHIFT itd.), dla których dodatkowo kontrolujemy ustawianie odpowiednich przerzutników. Etap następny to uruchomienie przyłącza magistrali — dokonujemy tego przy pomocy oscyloskopu kontrolując pojawianie się odpowiednich sygnałów. Następnie sprawdzamy działanie dekodera US14 wymuszając stany linii D4—D6 z pomocą przełączników, podobnie D0—D2 i linia STB. Przy prawidłowej pracy układu komputer odczytuje odpowiednie klawisze. Kontrolujemy też generowanie sygnałów CAPS i SYMBOL SHIFT oraz Q. Po uruchomieniu tej części można już włożyć w podstawkę EPROM-a i skontrolować poprawność pracy. W przypadku bezbłędnego jego zaprogramowania nie powinno być niespodzianek w działaniu klawiatury.

Całość należy odpowiednio obudować, najlepiej umieszczając klawiaturę w oddzielnej obudowie tak, aby można było ją przemieszczać po stole czy biurku. Nie zaleca się przy tym wydłużania przewodów przyłącza klawiatury ponad 50 cm. Szczegóły konstrukcyjne zależą od Czytelników.

Na zewnątrz należy zamontować wyłączniki blokowania klawiszy CAPS i SYMBOL SHIFT, automatycznego generowania EXT, wyłączania blokady Q (normalnie tryb ten powinien być zablokowany!) oraz przełączania trybu pracy klawiatury (zestawu oraz ułożenia klawiszy).

Wspomnieliśmy o możliwości wprowadzenia kilku trybów pracy klawiatury. Drugim trybem może być np. symulowanie klawiatury polskiej maszyny do pisania, z identycznym ułożeniem polskich liter (w wersji modelowej zamieniono między sobą już w trybie podstawowym litery Z i Y) oraz klawiszami funkcyjnymi z boku i na górze. O tym jednak w następnym odcinku, gdzie dodatkowo napiszemy o poprawieniu popularnego edytora TASWORD tak, aby przy wykorzystaniu opisanej klawiatury nie gubił znaków, przenosił polskie litery do następnego wiersza, realizował funkcję CAPS LOCK dla polskich liter itd.

Nasz test...

TIMEX 2048

Dzięki uprzejmości Dyrekcji CSH redakcja „MT" miała możliwość bliższego zapoznania się z mikrokomputerem TIMEX 2048 wraz z osprzętem. Mikrokomputer ten był sprzedawany przez Składnicę Harcerską w wytypowanych placówkach na terenie całego kraju.

TIMEX 2048 jest w naszym odczuciu mikrokomputerem wyłącznie amatorskim, zastosowaliśmy więc odpowiednie kryteria oceny. W ich świetle ocena mikrokomputera wypadła zupełnie pozytywnie.

W porównaniu z ZX Spectrum, z którym jest blisko spokrewniony, TIMEX 2048 wyróżnia się kilkoma istotnymi ulepszeniami. Obudowa nie tylko jest bardziej estetyczna ale i zapewnia elektronice lepsze warunki chłodzenia, co zapewne nie pozostanie bez wpływu na niezawodność. Klawiatura jest wygodniejsza i trwalsza nawet w zestawieniu ze Spectrum, mimo to jednak trudno uznać ją za odpowiednią do przetwarzania tekstów na większą skalę. W przypadku komputera domowego można się z tym jednak pogodzić. Wyłącznik napięcia zasilającego wróży długie życie gniazdu zasilania i zmniejsza ryzyko „zatkania" przetwornicy wskutek stanów nieustalonych. Wbudowany na stałe interfejs joysticka standardu Kempston zaoszczędzi telegraczom dodatkowych wydatków. Powiększony głośniczek nareszcie zasługuje na swoją nazwę. Wyjście monitorowe pozwala dołączyć mikrokomputer do standardowego monitora, np. typu Neptun 156, co zapewnia większą rozdzielczość, a w konsekwencji i jakość obrazu (szanujmy oczy!). Ta ostatnia zaleta nabiera znaczenia zwłaszcza w połączeniu z dodatkowymi trybami graficznymi.

W porównaniu ze Spectrum, TIMEX, dysponuje dodatkowym trybem graficznym o podwojonej rozdzielczości poziomej (512×196 punktów). Liczba możliwych do przedstawienia na ekranie barw redukuje się co prawda do dwóch, ale w zamian za to uzyskujemy możliwość czytelnego przedstawienia 64, a nawet 84 znaków w wierszu. Inny tryb graficzny dzieli ekran standardowo na siatkę 256×192 punktów, dając jednak podwyższoną rozdzielczość barwną. Każdej ósemce punktów ekranu, sąsiadujących w poziomie, można przypisać oddzielny bajt atrybutów. Tak więc możliwe jest np. wyświetlenie tekstu, w którym każda z ośmiu linii znaku ma inną barwę. Ten tryb udostępnia nowe

możliwości zwłaszcza w zakresie wielobarwnej grafiki, co ucieszy zapewne programistów gier. Nowe tryby graficzne wymagają oczywiście dodatkowej pamięci ekranu, a tę można oczywiście wykroić tylko kosztem obszaru RAM dla programu i danych. Obszar RAM liczy ogółem tylko 48 KB, każdy jego uszczerbek jest więc bolesny. Ponieważ BASIC jest identyczny jak w Spectrum, nie udostępnia on nowych możliwości graficznych TIMEX-a 2048. Pozostaje programowanie grafiki przy pomocy instrukcji POKE albo w innych językach, np. w asemblerze – całkiem jak w C-64.

Przejęcie zawartości ROM z ZX Spectrum ma jednak oczywistą zaletę: pozwala korzystać ze znacznej większości bogatego oprogramowania dla Spectrum. Drobne różnice w sposobie obsługi klawiatury sprawiają jednak, że ok. 10% programów dla ZX Spectrum nie działa na TIMEX-ie poprawnie. Na szczęście dotyczy to głównie gier. Nabywcy TIMEX-a mogą też oczywiście korzystać z obfitego już piśmiennictwa dotyczącego Spectrum.

W toku dwutygodniowej intensywnej eksploatacji TIMEX 2048 sprawował się bez zarzutu. Uparcie nasuwała się tylko jedna uwaga: dlaczego dodając interfejs Kempston nie wyposażono TIMEX-a od razu w interfejs RS232, chociażby z uproszczonym protokołem transmisji? Z technicznego punktu widzenia byłoby to do zrealizowania bez większych problemów, a dałoby komputer naprawdę samowystarczalny — przynajmniej w warunkach amatorskich. Tak czy inaczej nabycie TIMEX-a 2048 można polecać zarówno hobbistom, jak i niezamożnym klubom. Cena jest co prawda wyższa niż giełdowe notowania Spectrum, w zamian uzyskujemy jednak sprzęt wartościowszy i objęty gwarancją.

Ocena urządzeń peryferyjnych nie będzie już niestety tak pozytywna, jak samego komputera. Magnetofon cyfrowy TIMEX 2020 okazał się co prawda urządzeniem estetycznym, niezawodnym i łatwym w eksploatacji: wbudowany licznik taśmy i głośniczek pozwalały bez trudu lokalizować na taśmie poszczególne zbiory, zaś mikrofon objaśniać poszczególne zapisy słownymi komentarzami. Przy poszukiwaniu zbiorów o nieznanym położeniu na taśmie cenną pomocą był klawisz szybkiego przewijania w trakcie odtwarzania. Pasmo przenoszenia: 200...6300 Hz pozwala nie tylko na pewną pracę w standardowym trybie zapisu, ale i na stosowanie różnych „przyspieszaczy". Tych, którzy chcieliby ładny i poręczny magnetofon użyć w charakterze „walkmana",





czeka jednak rozczarowanie: kasowanie i zapis odbywają się prądem stałym, bez podkładu. Muzyka i głos są więc zapisywane ze zniekształceniami i przy znacznym poziomie szumów. Inna sprawa, że przyjęty sposób zapisu jest dogodniejszy dla sygnałów cyfrowych.

Istotną wadą magnetofonu jest bateryjne zasilanie. Cztery „paluszki" R-6 wyczerpują się szybko. Można co prawda dołączyć zewnętrzny zasilacz 6 V, oznacza to jednak nie tylko dodatkowy wydatek, ale i zagęszczanie sieci przewodów na stole. W sumie magnetofon jest więc nie tylko mniej uniwersalny, ale i znacznie kosztowniejszy od standardowych „kaseciaków" produkcji krajowej, które w roli pamięci masowej sprawują się także całkiem zadowalająco. Nabycie magnetofonu TIMEX 2020 jest opłacalne wtedy, gdy będzie on intensywnie eksploatowany — np. w klubie.

Pamięć na dyskach elastycznych do TIMEX-a 2048 należy uznać za kosztowne nieporozumienie. Udostępniony redakcji „MT" zestaw obejmował: dwie stacje dysków 3 cale, kontroler, zasilacz sieciowy i interfejs: w sumie pięć oddzielnych elementów. Wszystkie pudełka trzeba oczywiście połączyć przewodami. Zasilanie należy doprowadzić oddzielnie do obydwu stacji i kontrolera. Po zmontowaniu całość zajęła na biurku większą powierzchnię niż IBM PC. Przy pracy z dyskami całkowita rezygnacja z magnetofonu na ogół nie będzie możliwa, jako że większość oprogramowania jest jednak dostępna na kasetach, one też będą podstawowym nośnikiem do transferu zbiorów między użytkownikami TIMEX-ów i Spectrum. W efekcie na stole znalazło się osiem luzno ze sobą połączonych elementów, nie licząc wetkniętego do komputera interfejsu — w tym trzy zasilacze sieciowe. Abstrahując od sprawy estetyki, taki zestaw nie jest ani wygodny w obsłudze, ani tym bardziej niezawodny — wystarczy pomyśleć o mnogości złącz i narażonych na wyrwanie przewodów.

Kontroler dysków zawiera własny mikrokomputer z procesorem Z80, realizującym funkcje systemu operacyjnego. To dobre rozwiązanie, nie absorbujące procesora TIMEX-a 2048 i nie ograniczające pamięci operacyjnej. System operacyjny TOS (nie mylić z oprogramowaniem ATARI ST!) oferuje wiele zaawansowanych jak na komputer domowy właściwości, jak np. hierarchiczną, drzewiastą strukturę katalogu, możliwość ukrywania i ochrony zbiorów — jak np. MS-DOS PC-DOS. Obsługa systemu jest dość prosta — np. zapis i odczyt programów odbywa się dobrze znanymi instrukcjami SAVE i LOAD.

Czy rozbudowane mechanizmy TOS można efektywnie wykorzystać to inna sprawa. Producent podaje, że sformatowany dysk mieści na jednej stronie 160 KB na 40 ścieżkach. Warto jednak wiedzieć, że 16 KB zajmuje TOS, zapisujący się automatycznie podczas formatowania, a dalsze 4 KB — katalog dyskietki. Użytkownikowi pozostaje 140 KB, czyli mniej niż w w C-64. Mimo małej pojemności jednostka allokacji wynosi aż 1 KB. Oznacza to w praktyce, że przestrzeń dysku zajęta przez zbiór zaokrąglana jest w górę do pełnych kilobajtów. Jeśli zbiór liczy tylko 100 bajtów, TOS zarezerwuje dla niego 1 KB. Przy tak rozrzutnej gospodarce pamięcią i niewielkiej pojemności np. hierarchiczna struktura katalogu jest bardziej ciekawostką niż użytecznym narzędziem.

Najistotniejszym czynnikiem, rzutującym na użyteczność pamięci dyskowej, jest jednak fakt, iż stanowi ona twór obcy, nie uwzględniony pierwotnie w architekturze systemu ZX Spectrum TIMEX. Przytłaczająca większość oprogramowania przystosowana jest do współpracy wyłącznie z magnetofonem i bezpośrednio odwołuje się do odpowiednich programów w ROM.

Nie istnieje tak cenna w IBM-PC lub chociażby nawet w C-64 lub CPC możliwość „naginania" wektorów systemu operacyjnego, umieszczonych w pamięci RAM. Przełączenie z pamięci kasetowej na dyskową wymaga więc w przypadku Spectrum gruntownej niekiedy przeróbki programu. Do tego dochodzą trudności z przenoszeniem na dyski, a następnie uruchamianiem programów zapisanych pierwotnie na kasetach — zwłaszcza jeśli składają się one z kilku części. Stacja dysków nie zastąpi więc magnetofonu, a jej praktyczne zastosowanie ograniczy się głównie do programów tworzonych samodzielnie przez użytkownika. Jeżeli jednak decydując się na stację dysków musimy zrezygnować z głównego atutu Spectrum, jakim jest oprogramowanie, lepiej od razu wybrać inny komputer, z „prawdziwym" systemem operacyjnym — chociażby SVI-738, także rozprowadzany przez CSH. Trzeba też zaznaczyć, że dyskietki o średnicy 3 cali są gatunkiem egzotycznym i bez przyszłości stosowanym na większą skalę tylko w sprzęcie Amstrad-Schneider. Ich cena jest wyższa niż innych dysków, a możliwość transferu danych z innymi systemami — praktycznie żadna.

Kontroler dysków zawiera dodatkowo dwa interfejsy RS-232. Rozwiązanie zastosowane w Interface 1, znacznie wygodniejsze w użyciu, przypadło nam jednak zdecydowanie bardziej do gustu. Jeśli dodać, że Interface 1 posiada złącze typu „żeńskiego", zaś TIMEX — „męskiego", wskutek czego korzystanie z tego samego przewodu nie jest możliwe, otrzymamy pełny obraz istniejącego zamieszania.

Reasumując: TIMEX 2048 prezentuje się korzystnie jako prosty i tani komputer domowy, współpracujący z magnetofonem. Decydując się na jego zakup trzeba mieć pełną świadomość ograniczonych możliwości jego rozbudowy. Ten, kogo interesują w przyszłości poważniejsze zastosowania, a zwłaszcza pamięć masowa i interfejsy „z prawdziwego zdarzenia", niech od razu zainteresuje się innym sprzętem. Zaoszczędzi mu to nie tylko rozczarowań, ale i — *per saldo* — wydatków.

Osobnym problemem jest dokumentacja. Do testowanego zestawu była dołączona oryginalna dokumentacja angielska. W przypadku mikrokomputera była to zresztą dokumentacja nie do TIMEX-a, lecz do Spectrum. Dostawca komputera — firma POLBRIT — dołączyła tylko jedną, lakoniczną, dwustronicową, niechlujnie opracowaną ulotkę, wyjaśniającą pobieżnie różnice między Spectrum i TIMEX-em. Takie lekceważące podejście do klienta wystawia firmie POLBRIT jak najgorsze świadectwo. Wypada mieć nadzieję, że w przyszłości TIMEX-y otrzymają porządną dokumentację w języku polskim, zawierającą zarówno przykłady wykorzystania dodatkowych możliwości komputera, jak i listę bardziej popularnych programów dla ZX Spectrum nie pracujących na TIMEX-ie. Pozwoliłoby to uchronić użytkowników TIMEX-a przed nietrafionymi zakupami oprogramowania.

Roland Wacławek

SEMINARIUM „INFORMIKA"

ASEMBLER GENS 3

Część 1

Program Gens 3 firmy Hisoft jest asemblerem procesora Z80 przeznaczonym dla użytkowników ZX Spectrum 48 KB.

Ma postać relokowalnego kodu maszynowego i składa się funkcjonalnie z 2 części: edytora programu źródłowego i właściwego asemblera. Części te zostaną opisane oddzielnie, ponieważ opis edytora może być wykorzystany przy pracy z innymi translatorami firmy Hisoft (HP4S, C-Compiler).

Genes 3 jest przystosowany do współpracy z pamięcią masową w postaci taśmy magnetofonowej. Możliwość współpracy z pamięcią dyskową lub microdrive zostanie opisana dalej.

Genes 3 akceptuje programy źródłowe wygenerowane przez program monitora Mons 3 bezpośrednio z kodu binarnego.

Załadowanie i uruchomienie

CLEAR m: LOAD „Genes 3" CODE n (m < n)

Ze względu na maksymalne wykorzystanie pamięci dla programu źródłowego, który jest lokowany powyżej asemblera, stosujemy praktycznie np. m = 29999 n = 30000

a) Pierwsze uruchomienie:
RANDOMIZE USR n
Stosujemy tylko jeden raz!
Następne uruchomienia po powrocie do Basica:

b) Kasujące program źródłowy:
RANDOMIZE USR n + 56

c) Zachowujące program źródłowy:
RANDOMIZE USR n + 61

Po pierwszym uruchomieniu program pyta o rozmiar bufora. Należy wprowadzić liczbę od 0 do 9. Naciśnięcie tylko ENTER jest równoważne podaniu 4.

Bufor jest rezerwowany za kodem asemblera i jego wielkość jest określona wzorem: i * 256 bajtów dla i = 1,2...9 64 bajty dla i = 0

Bufor ten jest rezerwowany dla umożliwienia asemblacji programów źródłowych, które w całości nie mogły być umieszczone w pamięci operacyjnej (patrz zlecenie T edytora i dyrektywa F asemblera).

Asembler wykorzystuje następujące obszary pamięci operacyjnej:

Nazwa	adres początkowy
Kod asemblera	A = n
Bufor	B = n + 7238
Program źródłowy	C = n + 7238 + i * 256
Tablica symboli	D = PEEK (n + 54) + 256 * PEEK (n + 55)
Program wynikowy	E = D + dług. tabl. + 2

gdzie: n — adres początkowy ładowania
i — liczba bloków bufora

1) Rozmiar bufora dla danego programu źródłowego ładowanego z taśmy musi być identyczny z rozmiarem podanym w trakcie tworzenia tego programu.
2) Rozmiar tablicy symboli jest określany podczas asemblacji.
3) Umiejscowienie programu wynikowego można zmienić.

EDYTOR PROGRAMU ŹRÓDŁOWEGO

1. Format zleceń edytora

Edytor zgłasza się znakiem „ " oczekując na zlecenie. Działanie klawiszy funkcyjnych używanych przez edytor jest następujące:

ENTER — zakończenie wprowadzania (zlecenia lub linii)

EDIT — powrót do fazy oczekiwania na zlecenie

DELETE — skasowanie ostatnio napisanego znaku

Kursor w prawo — przejście do następnej pozycji tabulacji

Kursor w lewo — powrót do pierwszej pozycji tabulacji

Zlecenia edytora mają następujący format: A, m, n, s, t

A — litera oznaczająca zlecenie (mała lub duża)

m n — liczby całkowite z przedziału (1,32677)

s t — teksty o długości do 20 znaków

, — separator parametrów zlecenia

Uwagi:

4) Separator można zmienić na inny znak (patrz zlecenie S).
5) Spacje w zleceniu są ignorowane, za wyjątkiem spacji wewnątrz tekstu s lub t.
6) Każdy z parametrów może być pominięty. W pewnych zleceniach wartości te są wstępnie ustawione.
7) Błędnie napisane zlecenia są ignorowane (komunikat „Pardon?")

8) Ostatnio wykonane zlecenie jest pamiętane i może być wykonane powtórnie po naciśnięciu klawisza EDIT.

2. Wprowadzanie i redagowanie programu źródłowego

I m,n — automatyczna numeracja linii programu. (10,10)
m — numer pierwszej linii, n — krok

N m,n — renumeracja wszystkich linii programu.
m — nowy numer pierwszej linii, n — krok

D m,n - usuwanie linii programu. (Podać dwa parametry!)
m — pierwsza, n — ostatnia linia do usunięcia.

L m,n — listowanie programu. (1,32767)
m — pierwsza, n — ostatnia linia listowania

K n — liczba linii listowanych zleceniem L. (15)

M m,n — kopiowanie linii (Podać dwa parametry!)
m — istniejąca linia, n — linia utworzona

Uwagi:

9) Linia tekstu może zawierać do 64 znaków.

10) Pojedyncze linie można wprowadzać. Podając numer linii i tekst.

11) Wprowadzenie wyłącznie numeru linii bez tekstu powoduje kasowanie linii o tym numerze.

12) Kopiowanie linii m do istniejącej linii n powoduje kasowanie tekstu linii n. Linia m pozostaje w programie.

F m,n,s,t — poszukiwanie tekstu s w programie.
m — pierwsza, n — ostatnia przeszukiwana linia, t — tekst zastępujący s
Po znalezieniu tekstu w linii s edytor wykonuje zlecenie E x.

E m — edycja linii m dla dokonania zmian tekstu w linii. Działają następujące klawisze funkcyjne:

SPACE — przesuwa kursor o 1 pozycję w prawo,
DELETE — przesuwa kursor o 1 pozycję w lewo.
Kursor w prawo — przesuwa kursor do następnej pozycji tabulacji.
ENTER — wprowadzenie redagowanej linii ze zmianami do programu.

Po wykonaniu zlecenia E dostępne są następujące podzlecenia:

L – wyświetlenie redagowanej linii.
Q — zakończenie edycji z pominięciem dokonanych zmian.
R — ponowna edycja tej samej linii w wersji niezmienionej.
K — usunięcie jednego znaku na aktualnej pozycji kursora.
Z — usunięcie wszystkich znaków na prawo od pozycji kursora włącznie
X — ustawienie kursora na końcu linii i przejście do zlecenia I

I — wstawianie znaków począwszy od pozycji kursora.
DELETE — powoduje skasowanie ostatniego znaku
Kursor w prawo — przesunięcie kursora w prawo o 1 pozycję
ENTER — zakończenie wstawiania i powrót do zlecenia E.

C — zmiana znaków wskazywanych przez kursor.
DELETE — przesuwa kursor o 1 pozycję w lewo
ENTER — zakończenie zmian i powrót do zlecenia E.

S — wykonanie zlecenia F. Odnaleziony tekst s zostanie zastąpiony tekstem t, edytor przejdzie do poszukiwania następnej linii zawierającej tekst s.

F — odnaleziony tekst s nie zostanie zastąpiony tekstem t, edytor przejdzie do poszukiwania następnej linii zawierającej s.

3. Użycie pamięci taśmowej i drukarki

P m,n,s — nagranie programu źródłowego na taśmę.
m — pierwsza, n — ostatnia linia programu, s — nazwa zapisywana w etykiecie na taśmie

G ,,s — wczytanie programu źródłowego z taśmy.
s — nazwa etykiety programu

T m,n,s — nagranie programu źródłowego w postaci przystosowanej do asemblacji bezpośrednio z taśmy.

W m,n — wyprowadzenie programu źródłowego na drukarkę.
m — pierwsza, n — ostatnia linia
liczba jednorazowo wprowadzonych linii jak w zleceniu K.

Uwagi:

13) Etykieta programu źródłowego ma standardowy dla Spectrum format typu „CODE". Pierwszy bajt równy 3, następnie 10 znaków nazwy, dalej po dwa bajty na długość i adres początku ładowania.

14) Pominięcie parametru s w zleceniu G powoduje wczytanie pierwszego zbioru z taśmy.

15) Jeśli wykonamy zlecenie G mając już program źródłowy w pamięci, następny program zostanie dopisany za istniejącym a całość otrzyma nową numerację linii (od 1 z krokiem 1).

16) Zbiory utworzone zleceniem T nie mogą być ładowane zleceniem G.

17) W kompilatorze HP4S w formacie powyższego zlecenia jest używana w miejsce litery T litera W.

4. Inne zlecenia

C — zamiana wprowadzonego programu źródłowego na program źródłowy o standardowej dla Gens 3 postaci.

X — wyświetlenie adresów początku i końca programu źródłowego.

V — wyświetlenie ostatnio wprowadzonych parametrów zleceń.

S,, d — zmiana separatora parametrow zleceń.

B — powrót do Basica.

A — asemblacja programu źródłowego. (Patrz część III opisu).

Uwagi:

18) W standardowej postaci programu źródłowego spacje między pozycjami tabulacji są zastępowane 1 znakiem sterującym CHR$ (9).

19) Zamiana na taką postać np. dłuższego programu wygenerowanego przez Mons 3 może trwać kilkanascie minut.

20) Adres początku programu jest przy obranym buforze stały, a adres końca jest pamiętany w komórkach (n + 54) i (n + 55).

21) Nie należy zmieniać separatora na znak spacji, ponieważ zablokuje to wprowadzanie zleceń mających ponad 1 parametr.

22) W kompilatorze HP4S zlecenie C jest oznaczone literą O. Ponadto w obu kompilatorach (HP4S i C) odpowiednikiem asemblacji jest kompilacja realizowana zleceniem C.

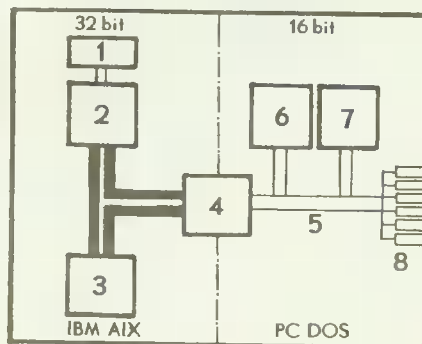
Tadeusz [illegible]

IBM RT/PC

Krótka, lecz jakże wymowna historia komputerów osobistych IBM PC, jest [illegible] już dobrze [illegible]. Komputer ten opracowany w ciągu zaledwie roku [illegible] [illegible] bardzo szybko światowym standardem sprzętu przeznaczonego dla indywidualnych stanowisk pracy. Był to jednocześnie pierwszy przypadek zastosowania przez [illegible] elementów standardowych (procesor Intel 8088), co, jak [illegible] dobrze wiadomo, znacznie ułatwiło kopiowanie komputera tego przez inne firmy. Konstrukcja ta jednak praktycznie [illegible] początku jej powstania [illegible] była nowoczesna – koncern [illegible] najchętniej wycofałby się z architektury systemu dość mocno ograniczającej [illegible]liwości rozwoju sprzętu. Pojawienie się modelu AT nieco poprawiło sytuację, [illegible] rosnące ciągle wymagania użytkowników stworzyły konieczność opracowania zupełnie nowego modelu.

Koncern [illegible] [illegible] mógł jednak i [illegible] chciał stworzyć komputera niezgodnego programowo z poprzednimi modelami. Zdecydowano [illegible] [illegible] krok kompromisowy i wykorzystano istniejący projekt procesora 32-bitowego (opracowany w IBM Labour Yorktown Heights pod roboczą [illegible] "projekt 801") wykorzystującego tzw. architekturę RISC [illegible] procesor stosowany w modelu AT. Celem tego kroku było po pierwsze wprowadzenie nowej generacji 32-bitowych PC, a [illegible] drugie – utrudnienie kopiowania wyrobu przez zastosowanie własnego, niestandardowego procesora.

Programową zgodność nowego komputera z poprzednimi modelami osiągnięto wprowadzając mieszaną architekturę 16- i 32-bitową – rys. 1. Część 32-bitową obsługuje procesor RISC (3) przy pomocy układu (2) inteligentnego wirtualnego zarządzania pamięcią (1). Część



Rys. 1. Struktura systemu [illegible] RT/PC.

ta jest połączona [illegible] fragmentem 16-bitowym komputera przy pomocy specjalnego konwertera kanałowego (4). [illegible] magistrali 16-bitowej (5) dołączony jest procesor [illegible] (6) [illegible] jego pamięć operacyjna (7) [illegible] zespół adapterów WE/WY (8). Dzięki takiej koncepcji możliwa jest równoległa praca [illegible] systemów operacyjnych [illegible] [illegible] AIX [illegible] PC DOS. Drugi [illegible] nich jest [illegible] bardzo dobrze znany. [illegible] [illegible] jest natomiast nowym wielodostępnym [illegible] wielozadaniowym systemem bazującym [illegible] UNIX-ie, lecz znacznie [illegible] niego szerszym. System ten pracuje [illegible] pamięci wirtualnej, przy czym komunikacja [illegible] sprzętem odbywa [illegible] przy pomocy warstwy systemu [illegible] oraz programu zarządzającego pamięcią wirtualną (VRM – Virtual Resource Manager). Konfiguracja taka umożliwia wykorzystanie bardzo bogatej biblioteki oprogramowania systemu [illegible] [illegible] nowych możliwości, stworzonych przez procesor RISC.

Kilka danych nowego modelu IBM RT/PC :

- jednoukładowy procesor o architekturze [illegible] posiadający [illegible] rejestrów 32-bitowych.
- kompatybilność [illegible] [illegible] IBM PC/AT osiągnięta przez zastosowanie koprocesora 80286 (plus [illegible] koprocesor [illegible]
- sprzętowe zarządzanie pamięcią :
  [illegible] TB (terabajt) pamięci wirtualnej
  [illegible] 4/16 [illegible] pamięci operacyjnej
  [illegible] sprzętowa ochrona stron pamięci
- magistrala [illegible] PC/AT [illegible] do 8 slotami
- możliwość wykorzystania kart IBM PC/AT
- mechanizmy dyskietek 5.25" o pojemności 1.2 [illegible]
- jednostki dysków typu Winchester [illegible] łącznej pojemności [illegible] 5,81 Gigabajta
- interfejs systemu graficznego IBM 5080
- ochrona danych
- [illegible] i kalendarz z zasilaniem bateryjnym
- interfejs myszy lub digitalizera
- [illegible] forma klawiatury

IBM PC/RT oferowany jest w [illegible] podstawowych wersjach :

- stołowej, oznaczonej IBM 6151 model 10
- wolnostojącej, oznaczonej IBM 6150 modele 20, 25 [illegible] A25, posiadającej [illegible] stosunku do wersji stołowej znacznie większe możliwości rozbudowy.

Scharakteryzujmy krótko konfigurację sprzętową komputera:

Procesor (3):
[illegible] to 32-bitowa jednostka zrealizowana [illegible] architekturze RISC (Reduced Instruction [illegible] Computer). Wykonuje 118 rozkazów 2- [illegible] 4-bajtowych. Szybkość pracy – od 1,6 do 2,1 MIPS przy [illegible] cyklu 170 [illegible]. Jest to mniej więcej [illegible] razy więcej, [illegible] dla IBM PC/AT, lecz mniej, [illegible] osiągają [illegible] modele wykorzystujące procesor Intel 80386. Najnowsze jednak wersje IBM 6150 model 115, 125 i [illegible] [illegible] już podwojoną szybkość pracy [illegible] [illegible] standardowo wyposażone w 32/64-bitowy koprocesor arytmetyczny zwiększający szybkość obliczeń zmiennoprzecinkowych ok. 40 razy.

Jednostka sterująca pamięci wirtualnej (2)
Jednostka [illegible] opracowana [illegible] postaci jednego układu scalonego posiada 40-bitowy [illegible] wirtualny, pozwalający zaadresować [illegible] Terabajt (') pamięci wirtualnej. Pamięć robocza jest [illegible] niemała – 16 MB. 32-bitowe [illegible] uzupełni[illegible] bi[illegible] kodu korekcyjnego, co pozwala skorygować wszystkie pojedyncze [illegible] wykryć wszystkie podwójne [illegible] większość potrójnych przekłamań.

Pamięć operacyjna (1):
[illegible] ona, [illegible] zależności od modelu, od [illegible] do 4 MB (w najnowszych modelach do [illegible] MB) pojemności. Koncepcja pamięci wirtualnej wymaga podziału [illegible] strony – [illegible] wielkość od 2 do 4 [illegible] [illegible] są [illegible] w całości przesyłane [illegible] pamięci zewnętrznej do operacyjnej. Mechanizm wymiany stron jest oczywiście niewidoczny dla użytkownika. Strony posiadają ochronę sprzętową.

Koprocesor IBM PC/AT (6):
Umożliwia on wykonywanie wszystkich programów pracujących pod systemem operacyjnym PC DOS, obsługuje również karty dołączone do magistrali 16-bitowej oraz własną pamięć operacyjną (7). Można także zainstalować koprocesor arytmetyczny 80287.

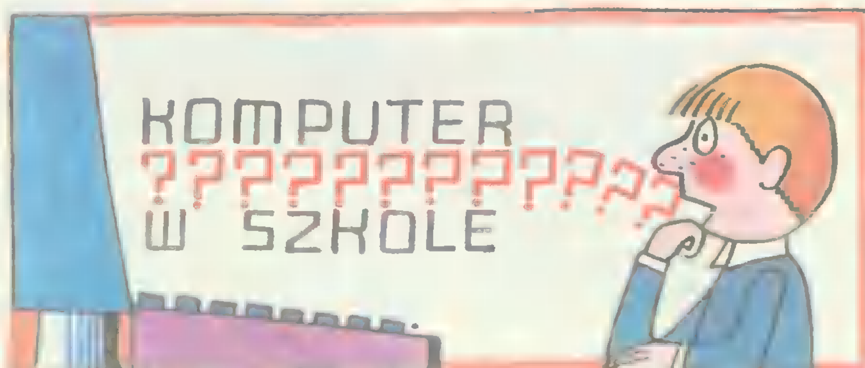
Magistrala systemowa:
Magistrala ta o szerokości 32 bity łączy wszystkie podstawowe elementy części 32-bitowej. Szybkość transmisji wynosi 23.5 MB/s.

Magistrala WE/WY (5):
Jest to typowa magistrala AT, pozwalająca wykorzystać wszystkie standardowe karty IBM.

Klawiatura:
[illegible] klawiatura posiada 102 klawisze, [illegible] tym 12 funkcyjnych. Rozłożenie klawiszy jest nieco inne, niż [illegible] poprzednich modelach.

Nowy [illegible] RT/PC przeznaczony jest przede wszystkim do pracy [illegible] systemach wymagających bardzo dużych mocy obliczeniowych, [illegible] zatem [illegible] wszelkiego rodzaju systemach CAD/CAM, sieciach komputerowych (system operacyjny IBM AIX jest przystosowany do pracy [illegible] kilku rodzajach sieci) i innych zastosowaniach naukowo-technicznych. Oferowane jest już bogate oprogramowanie wykorzystujące procesor RISC, zawierające bardzo dobre kompilatory, bazy danych, systemy CAD/CAM i wiele innych pakietów. Nowe modele RT/PC dorównują już szybkością komputerom z procesorem 386, pod względem wydajności pracy znacznie je jednak przewyższają – czy to wystarczy dla szerszego przyjęcia się nowego standardu, trudno d[illegible]siaj jeszcze przewidzieć.

(m.u.



JAK ZROBIĆ KOMPUTEROWY TEST?

Każdy poprawny test jest zbiorem kilkudziesięciu (minimum 30) odpowiednio skonstruowanych pytań i zadań, a jego wyniki ujmowane są liczbowo i a reguły podlegają opracowaniom statystycznym. Test stanowi więc obiektywną próbę porownywania pewnych cech ludzi.

Oczywiście najistotniejszą rolę odgrywa odpowiedni dobór pytań i zadań. Załóżmy jednak, że ten problem został już pokonany i zastanowmy się jakie warunki powinien spełniać program, aby komputerowa forma przeprowadzania testów stanowiła ułatwienie dla osoby prowadzącej i jednoczesnie była atrakcyjna dla odpowiadającego.

Pakiet życzeń odnośnie programu jest obszerny. Komputer powinien wyręczyć prowadzącego w żmudnych procesach poprawiania testu, jego oceny czy omawiania odpowiedzi na poszczególne pytania. Program, a ściślej zestaw pytań, powinien być łatwo modyfikowany, aby prowadzący w prosty sposób mógł zamienić któreś z pytań na inne, czy wręcz wymienić cały zestaw. Komputer powinien kontrolować czas udzielania odpowiedzi, a nie powinien w łatwy sposób dać się oszukać. Pełna realizacja tego ostatniego życzenia jest trudna do zrealizowania w BASICU.

A teraz spójrzmy od strony odpowiadającego. Pracuje on w stresie, w związku z czym należałoby w maksymalnym stopniu „ułatwić mu życie". Program powinien więc we wprowadzeniu wyjaśnić na czym polega test, jak kontaktować się z komputerem i przynajmniej na jednym przykładowym pytaniu dać możliwość przeprowadzenia prób. Jednocześnie przykład pozwoli odpowiadającemu zorientować się w rozkładzie treści na ekranie monitora.

Istotą testu jest fakt, iż nie ma konieczności odpowiadania kolejno na poszczególne pytania. Dowolne z pytań można opuścić i później, jeśli czasu wystarczy powrócić do niego. Program powinien to uwzględniać. Powinien również dawać możliwość dokonywania poprawek, po błędnym naciśnięciu klawisza komputera.

W ten sposób powstał zbiór warunków, które należy uwzględnić przy pisaniu programu. Dodatkowo, warto wyposażyć komputer w zestaw polskich liter, co znacznie ułatwi pracę odpowiadającemu.



Publikowany poniżej listing jest przykładem takiego programu napisanego w BASIC-u dla ZX SPECTRUM. Program zajmuje w pamięci około 5 KB i składa się z kilku zasadniczych bloków.

Blok wierszy 5÷25 wprowadza, przy pomocy grafiki użytkownika, dodatkowe polskie litery. I tak w trybie graficznym pod klawiszami: A, B, C, E, F, L, N, O, S znajdują się odpowiednio litery: Ą, Ź, Ć, Ę, Ż, Ł, Ń, Ó oraz Ś. Ponieważ instrukcja POKE 23658,8 z wiersza 30 ustawia komputer w trybie C (tzn. dużych liter), w grafice uwzględniono jedynie duże polskie litery. Dodatkowo pod klawiszami M i K znajdują się znaki ᵐ oraz ˢ, wykorzystywane do wyświetlania czasu.

Blok wierszy 30÷90 realizuje wprowadzenie oraz przykład. We wprowadzeniu zawarte są najistotniejsze uwagi o teście oraz zapoznanie z obsługą klawiatury. Wiersze 55 i 60 zmuszają do znalezienia klawisza DELETE, a wiersz 90 — klawisza ENTER. Dopiero po poprawnym naciśnięciu klawisza DELETE istnieje możliwość przejścia do przykładu, który ma na celu zapoznanie odpowiadającego z organizacją informacji wyświetlonej na ekranie monitora oraz ze sposobem udzielania odpowiedzi i jej poprawiania.

Przejście do właściwego testu następuje po:
- przekroczeniu czasu przeznaczonego do namysłu,
- trzykrotnym naciśnięciu ENTER — nieudzieleniu odpowiedzi,
- udzieleniu dowolnej odpowiedzi, zarówno poprawnej jak i błędnej.

Te same zasady obowiązują przy rozwiązywaniu zasadniczego testu, z tym, że czas, którego w przykładzie jest za dużo, musi teraz wystarczyć na udzielenie odpowiedzi na szereg pytań.

Po zakończeniu przykładu sterowanie przejmuje blok 100÷195, który organizuje przebieg testu. Pętla z wiersza 105 wyzerowuje w buforze drukarki niezbędną liczbę bajtów. Tam są przechowywane informacje o udzielanych na kolejne pytania odpowiedziach — czy były one poprawne, błędne, czy też któreś z pytań zostało pominięte. Następnie rozpoczyna się zasadnicza, podwójna pętla programu: FOR R = 1 TO P : FOR Y = 1 TO J. Wartość P, zadeklarowana w wierszu 35 jako 3, oznacza dopuszczalną liczbę powtórzeń poszczególnych pytań i może być przez prowadzącego test zmieniona, natomiast J jest liczbą pytań występujących w teście.

W wierszu 125 następuje skok do podprogramu (400÷420), którego zadaniem jest wypisywanie i rysowanie elementów poszczególnych pytań. Po powrocie z podprogramu rozpoczyna się blok (130÷190) składający odpowiedź O$ i analizujący jej poprawność.

Po zakończeniu udzielania odpowiedzi komputer przystępuje do opracowania wyników testu i wystawienia końcowej oceny — wiersze 250÷330.

Ostatni z bloków (350÷375) przeprowadza omówienie testu, polegające na porównaniu udzielonych odpowiedzi z odpowiedziami poprawnymi oraz zaprezentowaniu niezbędnych uzasadnień.

Program kończą cztery podprogramy, z których podprogram (400÷420) był już omawiany. W kolejnym (wiersze 450÷480) znajduje się mini-zegar odmierzający czas namysłu odpowiadającego. Dokładność zegara można zwiększyć poprawiając wartość stałej Q w wierszu 500. Podprogram 500 pozwala „wyzerować" mini-zegar. Ostatni z podprogramów (wiersze 550 i 565) służy do konstruowania poprawnych odpowiedzi i wyjaśnień dotyczących kolejnych pytań testu.

Omawiany program nie będzie realizowany. Zatrzyma się w wierszu 30 na instrukcji MERGE „PYTANIA", oczekując na dołączenie podprogramu zawierającego zbiór pytań, odpowiedzi i omówień. Warto zaznaczyć, że ten dodatkowy podprogram na ogół przewyższa objętością program główny.

Jak zbudować podprogram pytań, by współpracował on z programem głównym?

Po pierwsze: nie powinien on zawierać wierszy o numerach występujących w programie głównym (instrukcja MERGE!). W związku z tym zarezerwowano dla niego wiersze o numerach od 999 do 9999.

Po drugie: w wierszu o numerze 999 należy umieścić w deklaracji DATA trzy liczby: pierwsza z nich, to liczba pytań w teście, druga — czas przeznaczony na odpowiedź (wyrażony w minutach), natomiast trzecia powinna być liczbą znaków w najdłuższej z przewidywanych odpowiedzi. Ta ostatnia liczba potrzebna jest do zadeklarowania tablicy udzielanych odpowiedzi (DIM W$ (J, I) — w wierszu 30).

Po trzecie: należy pamiętać, że dla każdego z pytań zarezerwowano 100 wierszy. Oczywiście przy krótkich sformułowaniach nie będą one w pełni wyko-

```
   1 REM KOMPUTEROWY TEST
   5 BORDER [illegible] LET Q=USR "A"
  10 FOR N=1 TO 17 READ H LET
Q=Q+H
  15 FOR S=1 TO 3 READ A POKE
Q+S,A
  20 NEXT S NEXT N
  25 DATA [illegible] 126,56,[illegible]
[illegible] 21,68,126
[illegible] 126,0
[illegible] 124,0,5,[illegible],40
[illegible] 42,42,[illegible],12,0
0,4,[illegible],60,[illegible],24,60
,64
  30 POKE 23658,8 CLS PRINT A
T [illegible] FLASH 1 " WCZYTAJ BLOK PY
TAŃ " MERGE "PYTANIA" LET D$=C
HR$ 12 LET E$=CHR$ 13
  35 RESTORE 999 READ J READ T
 READ I DIM W$(J,I) LET P=3
LET Z=23300 LET U=0 LET I$="OD
POWIEDZI" LET J$=I$( TO 8)+"Ź"
  40 PRINT AT 0,5 "[illegible] - INI
[illegible] - TEST",TAB [illegible] TAB [illegible]
 TAB [illegible] -ŁODEGO [illegible] -[illegible]
A AT 4,4, NA ROZWIĄZANIE [illegible] MINI
TESTU, ZŁOŻONEGO Z [illegible] PYTAŃ,
POWINNO CI WYSTARCZYĆ [illegible] MINUT
 ODMIERZANYCH PRZEZ ZEGAR. [illegible]
  45 PRINT "CZA SIĘ WŁĄCZNIE [illegible]
$," POPRAWNE. ZA POdANIE [illegible]
 I$," NIE MA KARY. [illegible] JEŚLI N
IE POTRAFISZ UDZIELIĆ " I$ [illegible]
TÓREŚ Z PYTAŃ NA-CIŚNIJ ENTER [illegible]
NASTĄPI PRZEJŚCIE DO KOLEJNEGO P[illegible]
TANIA.JESLI CZASU WYSTARCZY, BEDZ
IE JESZCZE MOŻLI-WOŚĆ [illegible]
OTNEGO POWROTU DO TEGO PYTANIA.
  50 PRINT AT 17,4 "KAŻDORAZOWO
PO UDZIELENIU " I$, " RÓWNIEŻ NAC
ISNIJ ENTER." PRINT AT 19,4,"PO
PRAWEK W ",I$," MOŻNA DOKONYWAĆ
KLAWISZEM DELETE." [illegible] PO PRZEC
ZYTANIU NACIŚNIJ DELETE"
  55 IF INKEY$="" THEN GO TO 55
  60 IF INKEY$<>D$ THEN INPUT ""
 PRINT #0,TAB 4,"NIE! SPRÓBUJ
NACISNĄĆ " JEDNOCZEŚNIE CAPS SH
IFT I 0" GO TO 55
  65 RESTORE 1049 GO SUB 500 F
OR R=1 TO P LET Y=0 GO TO 125
  70 NEXT R GO TO 80
  75 PAUSE 140
  80 CLS PRINT AT 9,5, PAPER 4
" TERAZ SKONCENTRUJ SIĘ ",#0,TA
B 6 "I NACIŚNIJ ENTER " PRINT
AT 5,20," - " PAPER 6,B$, PAPE
R [illegible] AT 5,6,J$, IF O$=B$ THEN PR
INT " DOBRA" GO TO 90
  85 PRINT " ZŁA" READ B$ PRI
NT AT 1,0, PAPER 4,B$
  90 IF INKEY$<>E$ THEN GO TO 90
 100 GO SUB 500
 105 FOR N=Z TO Z+J POKE N,0
 110 NEXT N LET U=0
 115 FOR R=1 TO P
 120 FOR Y=1 TO J IF PEEK (Z+Y)
>0 THEN NEXT Y NEXT R GO TO 19
5
 125 GO SUB 400
 130 LET A$=INKEY$ IF A$="" THE
N GO SUB 450 GO TO 130
 135 BEEP .12,40 LET B=B+1 IF
B[illegible]*I THEN POKE Z+Y,1 GO TO 190
-120*(Y=0)
 140 IF A$=E$ AND O$<>"" THEN PO
KE Z+Y,1 GO TO 175
 145 IF A$=D$ AND O$<>"" THEN LE
T O$=O$( TO LEN O$-1) GO TO 170
 150 IF A$=E$ AND Y=0 THEN GO TO
 70
 155 IF A$=E$ THEN GO TO 1[illegible]0
 160 IF CODE A$>CODE "9" AND COD
E A$<CODE "A" OR CODE A$>CODE "Z
" OR CODE A$<CODE "0" THEN GO TO
 130
 165 LET O$=O$+A$
 170 PRINT PAPER 6,AT 19-G,19,O$
; PAPER 7," " GO TO 130
 175 LET U=U+1 IF O$=B$ THEN PO
KE Z+Y,2
 180 IF Y=0 THEN GO TO 75
 185 LET W$(Y)=O$
 190 IF U<>J THEN NEXT Y NEXT R
 PRINT AT 0,2, FLASH 1,"3-KROTN
IE POWTÓRZONO PYTANIA" GO TO 25
0
 195 CLS PRINT AT 9,6, FLASH 1
;" KONIEC TESTU "
 250 BEEP .5,9 PAUSE 140 LET A
=0 LET B=A
 255 FOR N=Z TO Z+J IF PEEK N=2
 THEN LET A=A+1
 260 IF PEEK N=1 THEN LET B=B+1
 265 LET C=J-A-B
 270 NEXT N LET S=59-S LET D=T
-M-1 IF S=60 THEN LET S=0 LET
D=D+1
 275 CLS PRINT AT 5,9, INVERSE
1," WYNIK TESTU " RESTORE 330
 280 FOR N=1 TO 4 LET E=VAL CHR
$ (64+N) READ A$ PRINT PAPER 4
,AT 6+N+N,5;A$+I$ TAB 22," - " P
APER 6,E,
 285 NEXT N
 290 FOR N=1 TO 6 READ B$ IF A
<.2*N*J THEN GO TO 300
 295 NEXT N
 300 PRINT PAPER 6,"[illegible]",S,"s",AT
16,10, FLASH 1,B$
 305 PRINT #0," KLAWISZ O - OMÓW
IENIE TESTU" " KLAWISZ P - POWTÓ
RZENIE TESTU" " KLAWISZ Z - ZMIA
NA PYTAŃ"
 310 IF INKEY$="" THEN GO TO 310
 315 IF INKEY$<>"O" AND INKEY$<>
"P" AND INKEY$<>"Z" THEN GO TO 3
10
 320 IF INKEY$="Z" THEN GO TO 30
 325 IF INKEY$="P" THEN CLS GO
TO 35
 330 DATA " DOBRE "," ZŁE "," BR
AK "," CZAS ","TRAGICZNIE","SŁAB
IUTKO","PRZECIĘTNIE","DOBRZE","B
ARDZO DOBRZE","DOSKONALE"
 350 LET H=-1
 355 FOR Y=1 TO J GO SUB 400 I
F W$(Y,1)=" " THEN LET W$(Y)="-"
 360 PRINT AT 19-G,0, PAPER 6,"T
WOJA ",AT 19-G,18, PAPER 7,W$(Y)
 365 PRINT AT 18-G,0, PAPER 4,"D
OBRA ODPOWIEDŹ ", PAPER 7,TAB 18
,B$ RESTORE X+99 LET B=32 GO
SUB 550 PRINT AT 1,0, PAPER 4,B
$ GO SUB X+50
 370 PAUSE 0
 375 NEXT Y INPUT "" GO TO 305
 400 LET X=1000+100*Y RESTORE X
+49 CLS READ G IF Y=0 THEN R
EAD B$
 405 PRINT INVERSE 1,AT 21-G*(H>
=0),0,TAB 11," PYTANIE NR ",Y,TA
B 32 IF H>0 THEN GO TO 415
 410 PRINT INVERSE 1,AT 21-G*(H>
=0),0;TAB 6,R;" RAZ" PRINT AT 0
,8, PAPER 6," CZAS ",M$,"m ",S
$,"s "
 415 IF Y<>0 THEN READ B
 420 GO SUB 550 READ C$ PRINT
#0,C$ GO SUB X LET O$="" PRIN
T AT 19-G,6, PAPER 6 "ODPOWIEDŹ
" LET B=0 RETURN
 450 LET H=H-1 LET M=INT (H/Q/6
0) IF M=VAL M$ THEN GO TO 465
 455 LET M$="" IF M<=9 THEN LET
 M$="0"
 460 LET M$=M$+STR$ M
 465 LET S=INT (H/Q-60*M) IF S=
VAL S$ THEN RETURN
 470 LET S$="" IF S<=9 THEN LET
 S$="0"
 475 LET S$=S$+STR$ S PRINT PAP
ER 6,AT 0,16,M$,AT 0,20;S$ IF H
=0 THEN PRINT FLASH 1,AT 0,[illegible]
 KONIEC CZASU " GO TO 250-175*
(Y=0)
 480 RETURN
 500 LET Q=5.4 LET M$="00" LET
 S$=M$ LET M=0 LET S=M LET H=
T*Q*60 RETURN
 550 IF B=0 THEN RETURN
 555 LET B$=""
 560 FOR N=1 TO B READ B LET B
$=B$+CHR$ (B+[illegible])
 565 NEXT N RETURN
 998 REM OPRACOWANIE [illegible]
```

```
9 REM  P Y T A N I A
999 DATA 10,12,7

1005 PRINT AT 9,8; PAPER 5; "ZABAWA(...)KŁODA",AT 3,11; FLASH 1;" PRZYKŁAD ": RETURN
1049 DATA 2,"BAL"," NAPISZ WYRAZ KTÓRY MA TO SAMO ZNACZENIE, CO SŁOWA UMIESZCZONE NA ZEWNĄTRZ NAWIASU "
1055 RETURN
1099 DATA " ZABAWA TO BAL,KŁODA TO TEŻ BAL "

1105 PRINT AT 9,8; PAPER 5;" 2 5 8 11 ..": RETURN
1149 DATA 6,2,5,8,"UZUPEŁNIJ CIĄG O KOLEJNĄ LICZBĘ "
1155 RETURN
1199 DATA 84,32,29,23,46,22,21,84,34,21,39,40,104,36,34,21,84,17,84,30,35,36,38,46,25,24,34,29,21,34,-1,7

1205 FOR N=1 TO 5: READ A$: PRINT AT 3+2*N,9; INVERSE 1;N; INVERSE 0; TAB 12;A$
1210 NEXT N: RETURN
1249 DATA 1,1,6,"PODAJ NUMER INTRUZA W TEJ GRUPIE WYRAZÓW ","ŚLEDŹ","WIELORYB","REKIN","BARRACUDA","DORSZ"
1255 RETURN
1299 DATA 84,43,29,25,32,35,38,45,38,84,40,35,84,39,39,21,31,1,35,35,39,46,40,21,84,40,35,84,38,40,22,45

1305 FOR N=1 TO 5: LET B=16*N: READ A$: IF LEN A$=1 THEN LET A$=" "+A$
1310 PRINT AT 2+2*N,9;A$+" ": PLOT 103,155-B: DRAW 38,B-48: DRAW 8,0
1315 NEXT N: READ A$: PRINT AT 8,19;A$: RETURN
1349 DATA 2,3,38,21,31," PODAJ WYRAZ KTÓRY DOPISANY DO KAŻDEJ Z LITER ORAZ GRUP LITER UTWORZY NOWE WYRAZY ","OR","B","U","BUT","...?"
1355 RETURN
1399 DATA 84,22,21,38,21,31,84,84,22,38,21,31,84,84,43,38,21,31,84,84,22,41,38,21,31,84,84,26,38,21,21,34

1405 READ C
1410 FOR N=1 TO 5: READ A,B: FOR D=A TO A+80 STEP 40: PLOT D,B: DRAW C,0: DRAW 0,-C: DRAW -C,0: DRAW 0,C: NEXT D: NEXT N
1415 PRINT AT 7,13;"1    2    3",AT 11,13;"4    5    6"
1420 FOR N=0 TO 4 STEP 2: PLOT 100+N,45: DRAW 0,110: NEXT N
1425 FOR N=1 TO 5: READ A,B,C,D: CIRCLE A,B,5: GO SUB 1440+D: READ A,B,C,D: PLOT A,B: DRAW -9,-9: DRAW 9,-9: DRAW 9,9: DRAW -9,9: LET B=B-9: GO SUB 1440+D: NEXT N
1430 FOR N=1 TO 4: READ A,B,C,D: PLOT A,B: DRAW 14,0: DRAW 0,-14: DRAW -14,0: DRAW 0,14: LET A=A+7: LET B=B-7: GO SUB 1440+D: NEXT N: RETURN
1443 FOR E=-1 TO 1: PLOT A,B: DRAW -C*E+2*C*(E=1),7*(E=0)-7*(E=2): NEXT E: RETURN
1446 FOR E=1 TO 4: PLOT A,B: DRAW C-2*C*(E=2),C-2*C*(E=1 OR E=3): NEXT E: RETURN
1449 DATA 1,1,7," PODAJ NUMER FIGURY KTÓRA POWINNA ZNALEŹĆ SIĘ NA WOLNYM MIEJSCU",24,16,152,16,112,16,72,144,152,144,72,102,140,7,3,20,149,3,3,100,100,5,5,68,109,8,3,68,60,5,5,20,69,6,3,156,60,5,3,196,69,3,3,236,60,5,5,196,140,4,4,61,147,6,6,21,107,6,6,157,147,6,6,229,147,6,3
1455 RETURN
1499 DATA 84,34,31,38,46,45,105,45,31,29,84,38,114,43,34,29,25,105,34,35,22,38,21,23,21,30,100,64,39,29,104,84

1505 PRINT AT 7,0; PAPER 5;" ZOTARAM  SATBURS  REWIC  MALOSE "
1510 RETURN
1549 DATA 2,5,31,21,32,35,39,25," PODAJ ZESTAW LITER,KTÓRY NIE TWORZY NAZWISKA SŁAWNEGO KOMPOZYTORA "
1555 RETURN
1599 DATA 34,33,35,45,21,38,40,64,84,39,40,28,21,41,38,39,84,34,42,25,35,24,29,84,84,33,21,32,35,[illegible],25,84

1605 PRINT AT 7,2; PAPER 5;" MAŁŻEŃSTWO POWODZI ZDROWIE "
1610 FOR N=1 TO 6: READ A$: PRINT AT 8+6*(N>3)+2*N,6+13*(N>3); INVERSE 1;N; INVERSE 0; TAB 8+13*(N>3);A$
1615 NEXT N: RETURN
1649 DATA 2,1,7," PODAJ NUMER WYRAZU,KTÓRY PASUJE DO TRZECH WYRAZÓW Z PIERWSZEGO RZĘDU ","WYKRZYK","POLE","STAN","CZŁOWIEK","WYDZIAŁ","POPRAWA"
1655 RETURN
1699 DATA 84,31,21,105,24,45,84,41,45,38,21,46,34,30,25,39,40,84,39,40,21,34,25,33,84,23,46,25,29,38,113,84

1705 FOR N=1 TO 3: READ A$: PRINT AT 3+3*N,7;A$: NEXT N
1710 FOR N=0 TO 3: PLOT 44,64+N*24: DRAW 168,0: NEXT N: DRAW 0,-72: PLOT 44,64: DRAW 0,72
1715 NEXT N: DRAW 0,-72: PLOT 44,64: DRAW 0,72: RETURN
1749 DATA 2,8,5,4," USTAW BRAKUJĄCĄ LICZBĘ ","7 ? 9 5","31 4 15 12","? 13 6 11"
1755 FOR N=1 TO 3: FOR E=1 TO 5: PRINT AT 3+3*N,5+5*E;: IF E=2 THEN PRINT "*": NEXT E
1760 PRINT "*": NEXT E: NEXT N
1770 RETURN
1799 DATA 84,34,84,84,34,84,84,57,31,-1,84,84,12,84,17,84,5,5,34,-1,84,5,4,84,84,84,84,84,84,64,84

1805 FOR N=1 TO 5: READ A$: PRINT AT 3+2*N,9; INVERSE 1;N; INVERSE 0; TAB 12;A$
1810 NEXT N: RETURN
1849 DATA 1,1,8," PODAJ NUMER ZESTAWU LITER KTÓRY NIE TWORZY NAZWY SAMOCHODU ","TAIF","SARNEY","DORF","MUTPIKS","BANTRAT"

1855 RETURN
1899 DATA 26,29,21,40,34,35,25,25,25,34,21,84,26,38,32,84,24,36,41,40,34,29,31,34,40,30,21,22,21,34,40

1905 READ A$: PRINT AT 6,9;A$
1910 READ A$: PRINT AT 8,7;A$
1915 RETURN
1949 DATA 1,7,30,25,46,23,35,35,35," JAKI WYRAZ POWINIEN ZNALEŹĆ SIĘ W NAWIASIE ","REINOWACJA(...)RTINO","RO(J..I..)OIZE"
1955 FOR N=0 TO 1
1960 FOR E=0 TO 1: PLOT 76+E*F+...*N,102: DRAW 80-16*E,0,PI
1965 NEXT E: NEXT N: RETURN
1999 DATA 22,38,21,31,41,30,100,23,35,84,32,23,40,25,38,15,84,35,100,84,26,22,35,31,34,34,21,42,22,21,30,41

2005 CIRCLE 131,92,50
2010 FOR N=1 TO 6: READ A,B,A$,C,E: PRINT AT A,B;A$: PLOT 131,92: DRAW C,E
2015 NEXT N: RETURN
2049 DATA 0,2,7,6," USTAW BRAKUJĄCĄ LICZBĘ ",6,18,"1",0,50,10,20,"3",43,25,14,16,"2",43,-25,14,14,"4",0,-50,10,12,"5",-40,-25,6,14,"?",-43,25
2055 RETURN
2099 DATA 34,5,-2,6,17,6,84,0,84,6,-2,6,17,6,84,0,84,6,-2,6,17,1,84,0,84,6,-2,12,17,6,84

9999 REM OPRACOWAŁ M.T.SZCZEPAŃSKI
```

rzystane. Numer wiersza, w którym rozpoczyna się informacja o danym pytaniu, określa wyrażenie: $x = 1000 + 100 \cdot Y$ (w wierszu 400), gdzie Y jest numerem tego pytania.

Po czwarte: zarezerwowane 100 wierszy należy podzielić na połowę. Wiersze od X do X + 48 powinny zawierać instrukcje dotyczące graficznej ilustracji pytania lub zadania, zaś w deklaracji DATA umieszczonej w wierszu X + 49 należy zamieścić kolejno: zmniejszoną o jeden liczbę linii, które na ekranie monitora zajmuje, sformułowane do danego problemu, pytanie. W następnej kolejności łańcuch odpowiedzi i łańcuch pytania. Po nich mogą się znajdować ewentualne inne dane wykorzystywane w bloku wierszy od X + 48. W drugiej połowie (wiersze od X + 50 do X + 98) powinny się znaleźć informacje graficzne dotyczące rozwiązania, a w deklaracji DATA umieszczonej w wierszu X + 99: treść omówienia plus ewentualnie dodatkowe dane.

Po piąte: bloki wierszy od X do X + 48 oraz od X + 50 do X + 99 są podprogramami wywoływanymi z programu głównego instrukcjami: GO SUB X (w wierszu 420) i GO SUB X + 50 (w wierszu 365) i w związku z tym muszą być zakończone instrukcjami RETURN.

Wreszcie **po szóste**: pytanie przykładowe należy umieścić w wierszach od 1000 do 1099, przestrzegając wyżej przyjętych zasad.

Drugi z załączonych listingów jest właśnie przykładem podprogramu pytań. Jego analiza powinna ułatwić budowanie własnych zestawów pytań. Uważny Czytelnik zapewne dostrzegł różnicę między konstrukcją pytania przykładowego a zestawem pozostałych pytań. Wynika ona z dwóch przyczyn. Program nie posiada zabezpieczenia przed wścibskim okiem rozwiązującego, który może go przerwać przy pomocy BREAK-a. W związku z tym poprawne odpowiedzi i omówienie pytań zostały zakodowane. Odkodowanie ich odbywa się w podprogramie 500. Drugim powodem utajnienia odpowiedzi była chęć stworzenia możliwości rozwiązywania tego testu przez Czytelnika, który zdecyduje się wprowadzić do komputera te ponad 8 KB danych. Oba zamieszczone listingi zajmują po 9 ekranów na monitorze i wydrukowane są w liniach liczących po 32 znaki. Być może ułatwi to wprowadzanie danych, ponieważ ekran monitora i listing powinny być identyczne. Nietrudno zauważyć, że drugi z listingów zajmuje znacznie więcej pamięci. Jest to spowodowane zakodowaniem odpowiedzi. Wprawdzie istnieje możliwość schowania większości z tych liczb do pojedynczych komórek pamięci, ale i tak listing powinien zawierać je w formie jawnej.

Przy proponowanej konstrukcji programu test może składać się z 89 zadań. Jest to liczba absolutnie wystarczająca, choć nie jest żadnym kresem możliwości. Bardziej istotne ograniczenie stanowi pojemność pamięci komputera. Przeznaczając na jedno pytanie średnio 0,5 KB, będziemy w stanie zmieścić w pamięci SPECTRUM około 70 pytań.

Zamieszczony obok listing podprogramu pytań został przygotowany na podstawie 10 zadań zaczerpniętych z TESTU INTELIGENCJI (tzw. test IQ). Miejmy nadzieję, że psychologowie nie poczują się dotknięci faktem opublikowania fragmentów jednego z testów zdolności ogólnych, tym bardziej że pełny test wraz z odpowiedziami i omówieniem zamieściła swego czasu „WIEDZA I ŻYCIE". Na podstawie zaledwie 10 pytań nie można i nie wolno określać np. ilorazu inteligancji, czy wyciągać jakichkolwiek innych wniosków. Jest to za mała próbka, w dodatku nie standaryzowana i nie normalizowana. W naszym przypadku posłużyła jedynie jako ilustracja omawianej tematyki a dla Czytelników lubiących myślenie może stanowić przyjemną rozrywkę.

W praktyce szkolnej komputerowy test można wykorzystać jako atrakcyjny sprawdzian wiadomości, rzadziej jako sprawdzian umiejętności. Stosunkowo łatwo można napisać test wyboru i test uzupełnień. Podczas pisania należy zwracać szczególną uwagę na sformułowania pytań, aby odpowiedzi były jednoznaczne i w miarę krótkie. Jest oczywiste, że efektywność i efektowność testu zależy głównie od pomysłowości autora, jego umiejętności programowania oraz od możliwości komputera.

Pewnej modyfikacji wymagałby program główny, jeśli miałby dopuszczać odpowiedzi alternatywne, ale o tym problemie przy innej okazji. A może Czytelnicy zechcą podzielić się z nami swoimi uwagami i przemyśleniami?

Marek T. Szczepański

STACJA DYSKÓW ELASTYCZNYCH

W ostatnich latach wyraźnie wzrosło zapotrzebowanie na tanią i niezawodną pamięć zewnętrzną. Jedną z bardziej atrakcyjnych ofert jest dysk elastyczny: niewielki czas dostępu do informacji, duża szybkość transmisji dochodząca do 250 kBit/sek., łatwość wymiany nośnika informacji (dyskietki) oraz wysoka niezawodność, charakteryzująca się niewielką liczbą przekłamań (rzędu 1 na 10^8 przeczytanych bitów) oraz duża niezawodność konstrukcji mechanicznej: 1 błąd pozycjonowania na 10^6 ustawień głowicy.

Wszystkie te parametry sprawiły, że miękkie dyski stały się najpopularniejszą w zastosowaniach półprofesjonalnych pamięcią zewnętrzną. Na jednej dyskietce o średnicy 5 1/4 cala znajduje się od 35 do 192 ścieżek podzielonych na sektory. Mechanizm napędu dysków pozwala na jedno- lub dwustronny zapis dyskietki.

Jednym z najważniejszych elementów dysku jest głowica. Jej jakość rzutuje w głównej mierze na jakość zapisu i odczytu informacji na nośniku magnetycznym. W chwili obecnej rdzenie głowic wykonuje się najczęściej ze stopów metali, którym w procesie obróbki termicznej nadaje się amorficzną strukturę szkła. Stopy te nazywane są szkłami metalicznymi i posiadają doskonałe parametry magnetyczne, zarówno statyczne, jak i dynamiczne. Proces produkcji opiera się na wyrzucaniu przez dyszę stopu zawierającego Ni, Fe, Co, Al oraz inne składniki na wirujący walec miedziany. Osiąga się w ten sposób bardzo duże szybkości schładzania, zapewniające uzyskanie niezbędnej struktury. Tak otrzymane taśmy o szerokości kilku milimetrów służą do wytwarzania rdzeni głowic.

Szerokość ścieżek nie przekracza 0,3 mm — konieczne jest więc bardzo precyzyjne pozycjonowanie głowicy. Błędy jej ustawienia mogą powodować znaczne obniżenie poziomu sygnału. Mogą się pojawiać także przesłuchy z innej ścieżki. Mechanizm pozycjonowania musi zapewniać minimalną bezwładność, konieczną do uzyskania znacznych prędkości przesuwania głowicy w trakcie wyszukiwania ścieżki (czas przesunięcia głowicy ze ścieżki na ścieżkę wynosi około 30 ms). Obecnie stosuje się dwa rodzaje mechanizmów przesuwania głowic: mechanizm śrubowy oraz mechanizm wykorzystujący spiralę Archimedesa. Podstawową częścią obu tych mechanizmów jest wielobiegunowy silnik krokowy, zapewniający minimalny skok (obrót) o kąt od kilku do kilkunastu stopni. Silnik ten w przypadku mechanizmu śrubowego ma zamocowaną na osi metalową śrubę z gwintem trapezowym, po której ślizga się teflonowy wodzik (również o przekroju trapezowym), dociskany do gwintu śruby sprężyście — w ten sposób kasuje się luzy i eliminuje histerezę mechaniczną.

Zamiast śruby na osi silnika może być zamocowana okrągła teflonowa tarcza. W tarczy wyfrezowany jest żłobek o kształcie zbliżonym do spirali Archimedesa. Żłobek posiada profil trójkątny, a współpracuje z nim wodzik w postaci kulki metalowej, dociskanej odpowiednią sprężyną. Dzięki takiemu rozwiązaniu również i w tym układzie udało się wyeliminować luzy. Możliwa była także minimalizacja błędu wynikającego z niedokładności pozycjonowania wału samego silnika. Uzyskuje się ją poprzez zastąpienie spirali Archimedesa linią aproksymującą ją, złożoną z wycinków okręgów o kącie równym kątowi niedokładności pozycjonowania wału silnika oraz łączących je odcinków spirali. Dzięki temu uzyskano zamianę obrotowego ruchu silnika na skwantowany, posuwisty ruch głowicy.

Wstępna regulacja ustawienia głowicy dokładnie nad ścieżką realizowana jest w obu mechanizmach poprzez obracanie całego silnika względem chasiss. Powoduje to zmianę wyjściowego położenia śruby lub tarczy. W niektórych bardziej precyzyjnych mechanizmach stosuje się metody elektronicznego śledzenia ścieżki, oparte na zasadzie porównywania sygnału z głowicy o dwóch śladach, czytającej tą samą ścieżkę. Każde zejście ze ścieżki zmniejsza amplitudę sygnału w jednej połówce głowicy, co sygnalizuje układowi śledzącemu konieczność korekcji położenia głowicy.

Dla uniknięcia oczekiwania na rozpędzenie się dyskietki przy krótkich przerwach w jej czytaniu nie przerywa się pracy głównego silnika krokowego napędzającego wrzeciono, w którym jest zamocowana dyskietka. Dyskietka wiruje z prędkością 300 obr./min. i mogłaby być w tym czasie nadmiernie ścierana przez głowicę dociskaną bezpośrednio do nośnika magnetycznego. Unika się tego zjawiska stosując elektromagnetyczny układ podnoszenia (dociskania) głowicy. W nowoczesnych stacjach z głowicami o wysokiej gładkości, podnoszenie głowic nie jest potrzebne. W czasie dłuższych przerw wyłącza się również silnik główny tak, by powierzchnia dyskietki nie wycierała się o kopertę ochronną.

Wrzeciono mocujące dyskietkę jest łożyskowane za pomocą dwóch łożysk kulkowych. Przeniesienie napędu uzyskuje się za pomocą specjalnego zbrojonego paska, dzięki czemu drgania silnika nie są przenoszone bezpośrednio na dyskietkę. Stabilność obrotów jest zapewniona dzięki zastosowaniu 4-fazowego silnika krokowego zasilanego z generatora, którego częstotliwość jest stabilizowana kwarcem.

Należy zauważyć, że w punktach dyskietki różnie oddalonych od jej osi obrotu zmienia się prędkość nośnika informacji względem głowicy, czyli tym samym długość fali zapisywanej na magnetycznym podłożu. Powoduje to w niektórych gorszych jednostkach kłopoty z odczytem wewnętrznych, bądź zewnętrznych ścieżek dyskietki (mają tu swój udział także błędy pozycjonowania głowicy), uniemożliwiające maksymalne wykorzystanie dyskietki. Radą na ten mankament jest stosowanie odpowiednio regulowanej charakterystyki wzmacniaczy odczytu względnie zmniejszenie liczby sektorów przy dochodzeniu do środka dyskietki (to rozwiązanie jest stosowane znacznie rzadziej) np. w stacji Commodore VC 1541).

Chassis jednostki jest wykonane w postaci wtryskowego odlewu ze stopów lekkich — zapewnia to konstrukcji odpowiednią sztywność i niewielką masę.

Związany z mechanizmem układ elektroniczny zawiera wzmacniacz zapisu i odczytu wraz z układem formowania sygnału, generator zasilający główny silnik napędowy oraz układ sterujący krokowym silnikiem mechanizmu przesuwającego głowicę. O sposobach kodowania zapisywanej informacji napiszemy w następnym numerze.

„Młody Technik — InforMik" wydaje Instytut Wydawniczy „Nasza Księgarnia"

Rada Redakcyjna: doc. dr Zygmunt Dąbrowski, inż. Jerzy Jasiuk, dr Zygmunt Kalisz, mgr Zbigniew Słowiński, mgr inż. Jerzy Siek, dr Zbigniew Płochocki, Piotr Postawka, mgr inż. Roland Wacławek, prof. dr hab. Andrzej K. Wróblewski (przewodniczący), mgr inż. Grzegorz Zalot.

Zespół redakcyjny: „InforMik" redaguje zespół „Młodego Technika". Jerzy Klawiński (red. odpowiedzialny), Lidia Sadowska-Szlaga (red. techn.), Józef Trzionka (redaktor naczelny), Roland Wacławek (software), Grzegorz Zalot (hardware).

Stali współpracownicy: Wojciech Apel, Tadeusz Basisla, Jacek Jędrzejowski, Piotr Postawka, Marek Szczepański, Krzysztof Wiśniewski.

Adres redakcji: ul. Spasowskiego 4, 00-389 Warszawa, lub skr. poczt. 380, 00-950 Warszawa. **Telefony:** centrala: 26-24-31 do 36. Dział Łączności z Czytelnikami — wewn. 60, pozostałe działy: wewn. 42 i 47. Redaktor naczelny: 26-26-27 lub wewn. 87.

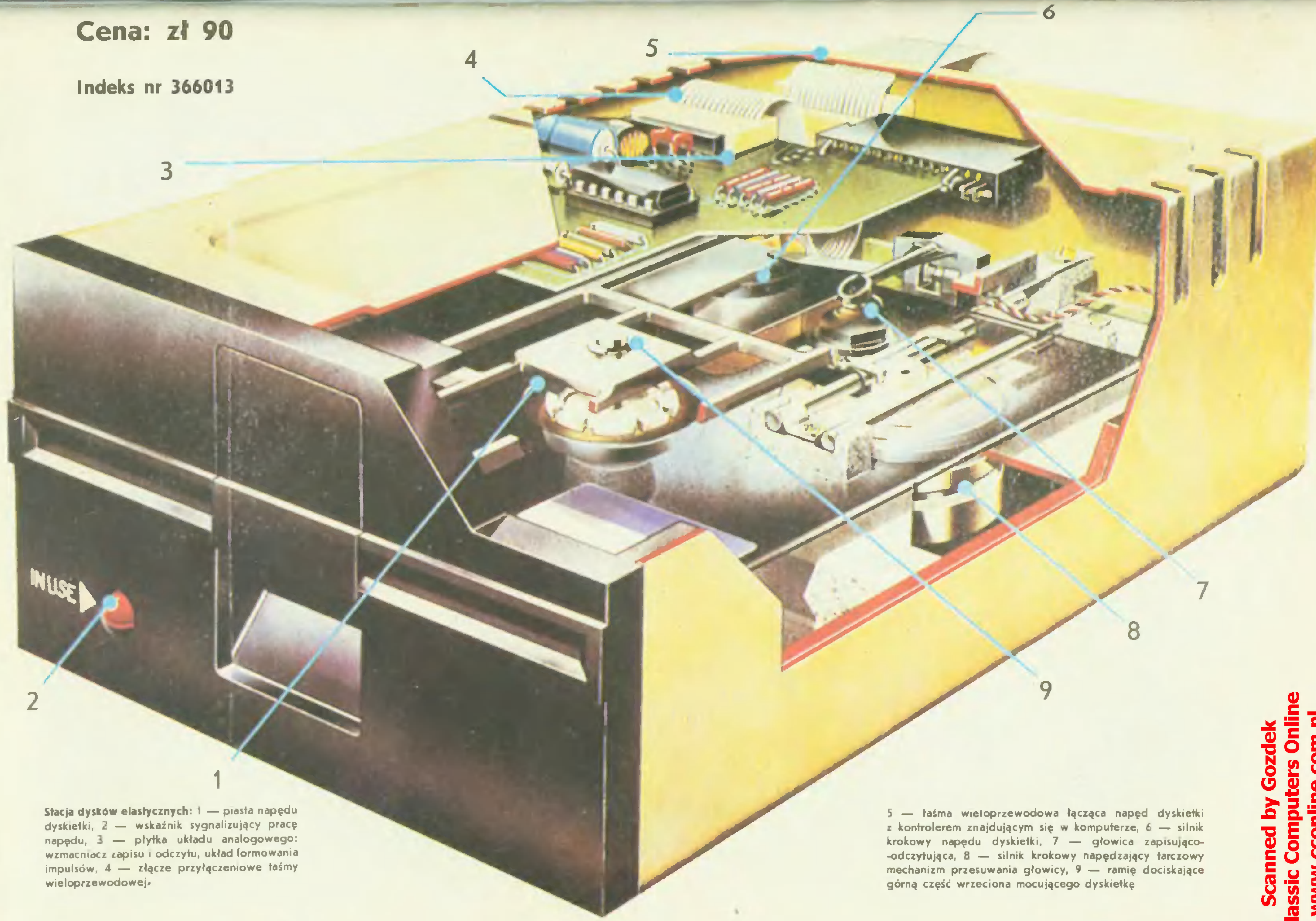
Warunki prenumeraty: ogólnie obowiązujące w kraju. **W STAŁEJ SPRZEDAŻY INFORMIK JEST W SALONIE WYDAWNICZYM „NASZEJ KSIĘGARNI"** ul. Spasowskiego 4A.

Redakcja zastrzega sobie prawo adiustacji i skracania nadesłanych materiałów. Artykułów nie zamówionych redakcja nie zwraca.

Druk: Zakłady Graficzne w Katowicach. Zam. 0450/1333/7
Nakład 100 315 egz. K-82

Cena: zł 90

Indeks nr 366013



Stacja dysków elastycznych: 1 — piasta napędu dyskietki, 2 — wskaźnik sygnalizujący pracę napędu, 3 — płytka układu analogowego: wzmacniacz zapisu i odczytu, układ formowania impulsów, 4 — złącze przyłączeniowe taśmy wieloprzewodowej, 5 — taśma wieloprzewodowa łącząca napęd dyskietki z kontrolerem znajdującym się w komputerze, 6 — silnik krokowy napędu dyskietki, 7 — głowica zapisująco-odczytująca, 8 — silnik krokowy napędzający tarczowy mechanizm przesuwania głowicy, 9 — ramię dociskające górną część wrzeciona mocującego dyskietkę